TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

VIVEK DEIVASIGAMANI

COMMUNICATION INFRASTRUCTURE ENABLING PARTICIPATION OF SOA DRIVEN MANUFACTURING ENTERPRISES IN DEMAND RESPONSE PROGRAMS

Master of Science Thesis

Examiner: Prof. Jose L. Martinez Lastra
Examiner and topic approved by the Council meeting of the Faculty of Engineering
Sciences on 5th October 2015

ABSTRACT

VIVEK DEIVASIGAMANI

Tampere University of technology
Master of Science Thesis, 66 pages, 25 Appendix pages
November 2018
Master's Degree Programme in Automation Engineering
Major: Factory Automation and Industrial Informatics
Minor: Production Engineering
Examiner: Prof. Jose L. Martinez Lastra

Keywords: energy management, demand side management, demand response, OASIS energy interop, OpenADR, web portal, portlet.

Energy is a necessary element for all industrial process and the demand for energy is inevitable. The difference between energy use and energy consumption focus the industrial users to decrease their baseline energy consumption based on their demand.

Demand Response program is one such technique which allows the industrial consumers to curtail the energy consumption from their normal energy usage. Therefore, explicating a communication infrastructure for demand response program will allow the manufacturing industries for seamless communication between the demand response program participants. This in turn helps the production planner to optimize their production plan accordingly, thereby reducing the total electricity cost for the manufacturing facilities and optimize their productivity.

At first, this paper describes the available Demand Response standards and harmonization between them. Secondly, a model for DR participants will be done followed by a layered approach for constructing the communication infrastructure. On the other hand, the information to be exchanged and protocols used for exchanging this information between the DR participants will be discussed. Finally, the developed communication infrastructure would be demonstrated in a conceptual platform.

PREFACE

வாழ்க வளமுடன் ! – Be Blessed !

First I would like to thank my Dad, Mom and My Brother who are the soul for me and keeping faith in me all the time. I thank all my friend back in India who motivated me to pursue a master’s degree.

I am grateful to Tampere University of Technology and Prof. Jose L. Martinez Lastra for accepting me to the degree program. My special thanks to Andrei Lobov, Anne Florea and others who were working with me in FastLab during my tenure.

Next I would like to thank my friends – Balaji, Naveen, Nirmal, Sriram, Sathish, Karthick, Prasanth, Sagar, Shivakumar, Anand who motivated me all the time while writing the thesis. I would also like to thank all my Bremen friends who supported me during my exchange period.

I am fortunate to have Danny, Santhosh and Iida who stood with me during all my up’s and downs.

Finally I would like to thank my employer Visual Components for their constant support.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

AHP	Analytic Hierarchy Process
API	Application Program Interface
ASHRAE	American Society of Heating, Refrigerating and Air-Conditioning Engineers
BACS	Building Automation and Control Systems
BEMS	Building Energy Management System
CDF	Controllable Device First
CIM	Common Information Model
CoAP	Constrained Application Protocol
CPP	Critical Peak Pricing
CRUD	Create, Read, Update and Delete
DB	Demand Bidding/Buyback
DOE	Department of Energy
DLC	Direct Load Control
DR	Demand Response
DRA	Demand Response Aggregator
DRAS	Demand Response Automation Server
DSRA	Demand Side Response Aggregator
DSM	Demand Side Management
DSO	Distributed System Operator
EC	Electricity Consumers
EDRP	Emergency DR Program
EM	Energy Market
EMIX	Energy Market Information Exchange
EMS	Energy Management Systems
FAST-Lab	Factory Automation Systems and Technologies Laboratory
FEMS	Factory Energy Management System
FO	Facility Operator
HP	Hourly Pricing
HTTP	Hyper Text Transfer Protocol
HVAC	Heating, Ventilating, Air-Conditioning
I/C	Interruptible/Curtailable
ICT	Information and Communication Technology
IDE	Integrated Development Environment
IoT	Internet of Things
IP	Internet Protocol
ISO	Independent System Operator
JAR	Java Archive
LCO	Load Control Object
LLC	Logical Link Control
MAC	Media Access Control
MAS	Multi-Agent System
MCP	Market Clearing Price
MES	Manufacturing Execution System
NIST	National Institute of Standards and Technology

OASIS	Organization for the Advancement of Structured Information Standards
OASIS EIOp	OASIS Energy Interoperation standard
ORM	Object-Relation Mapping
OSI	Open System Interconnection
QoS	Quality of Service
POM	Project Object Model
REMS	Residential Energy Management System
REST	Representational State Transfer
RFC	Request For Comments
RPC	Remote Procedure Calls
RTO	Regional Transmission Organization
RTP	Real-Time Pricing
SCF	Shiftable Class First
SEP	Smart Energy Profile
STN	State-Task Network
TCP	Transmission Control Protocol
TSO	Transmission System Operator
TOU	Time of Use
UDP	User Datagram Protocol
URL	Uniform Resource Locator
USP	Utility Service Provider
VEN	Virtual End Node
VTN	Virtual Top Node
XML	eXtensible Markup Language
XMPP	eXtensible Messaging and Presence Protocol
XSD	XML Schema Definition
W3C	World Wide Web Consortium

1. INTRODUCTION

In Manufacturing industry, transmute of raw materials into final product consumes electricity because of the machine drives[1]. The electricity consumption of these machine drives accounts half of the manufacturing industry's delivered electricity [1]. It is therefore responsibility of the manufacturing industries to be conscious on reducing their electricity consumptions. Reducing the electricity consumption in manufacturing industries without affecting the normal production schedule has been a principal area of research for past decade. Various energy consumption reduction technologies (system and simulation approach, integrating of smart meters) and related policies in manufacturing industries were proposed [2]. It is anticipated that cost-based load curtailment technique is acclaimed predominantly by the industrial consumers. Therefore, implementing Demand Response (DR), a category in Energy Management would be a suitable technique. According to Federal Energy Regulatory Commission [3] Demand response (DR), an energy consumption reduction technique, can be defined as "Changes in electric usage by demand-side resources from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized". Demand Response have a key feature of load curtailment technique based on the hourly electricity price or incentive for curtailing load during peak hours. Demand response can also be referred as Demand Side Management (DSM) strategy.

For initiating a demand response program, communication between the supply and demand sides is essential. The thesis work focuses on developing demand response communication infrastructure in manufacturing facilities. The first chapter begins with the problem definition, objective, assumption and limitations. The second chapter illustrates the background of demand response, standards related to demand response, and available communication protocols. The third chapter presents methodology to attain the objective of the thesis. The fourth chapter explains the communication architecture of demand response and illustrated with a manufacturing facility testbed. The fifth and the sixth chapters discuss the result, conclusion and future work which can be applied to demand response communication infrastructure.

[1] Machine drives are electric motors, pumps and fans in an electrical device

1.1 Thesis Background

In the manufacturing facilities, baseline energy consumption can be achieved by production planning of Manufacturing Execution System (MES) and Energy Management Systems (EMS). The effectiveness of efficient production planning and scheduling in manufacturing industries for reducing energy consumption were explained in [4]. The energy consumption in the manufacturing facilities can be reduced beyond the baseline by incorporating demand response program into the Energy Management System. Demand Response program encourages the manufacturing industries to reduce their baseline energy consumption during the off-peak hours. Furthermore, the program indirectly enhances scheduling (assign machinery and human resources) of manufacturing industries.

Previous research [5–7] in DR program recognized low level of participation was due to asymmetries in information and information exchange between the DR program participants. Therefore, a bidirectional communication can enable the information exchange between the demand response participants effectively. To achieve a high quality of service, the communication between the participants of DR program must be made explicit, easily accessible and efficient. For this reason, the transmission of information between DR participants such as the energy market context, the utility information system, consumer's energy management system should be effectively designed.

With the service-oriented architecture capability and available demand response standards discussed in chapter 2.2, the communication bottleneck between the supply and demand side participants in demand response program can be eliminated. Even though there exist different demand response standards, an outline for harmonizing the standards is needed for individual facility. This works facilitates an approach to model the demand response communication infrastructure with the DR standards and communication protocol. Furthermore, a web portal architecture is proposed for integrating the different demand response services provided between the DR participants in the communication infrastructure.

1.2 Problem Definition

Even though the industrial consumer's shows favor for adapting Demand Response program in their manufacturing facility, there exist challenges on the deployment of these programs. This is due to the lack of demand response communication infrastructure between the electricity supply utilities and industrial facilities. A generic, flexible, harmonized open communication infrastructure is missing in the research and deployment of demand response programs in the industrial facility.

1.2.1 Justification of work

Traditionally supply side is deterministic where the electricity providers predict and schedules the facilities load as "given quantity" to generate, distribute and transmit electricity.

Since the renewable energy are now-a-days used as additional supply resources to fossil fuels which are distributed and intermittent, it is hard for the electricity providers to predict and control the supply. Therefore, a demand side management infrastructure will help the electricity providers to determine the facilities electricity consumption for a given period. As facility operators opts for a transparent communication in the said infrastructure, a communication model to define the demand response program is required.

The role of actor's, demand response service provided to the actors, and information flow between the actor's must be identified to define the communication infrastructure for demand response. Furthermore, the demand response service must be integrated to the presentation layer that enables visibility of demand response events to facility operators.

1.2.2 Problem Statement

The manufacturing facilities participating in demand response program varies widely depending on the resources in their factory floor. A generic infrastructure could be achieved by identifying, categorizing, and sharing the demand response information between the demand response participants. The demand response information is available as a demand response open standard. This information can be identified and narrow tailored for the specific facility. The communication of the tailored information can be achieved with web service using communication protocols. Though the statement looks easily achievable, the following questions are to be defined and answered:

- What are the types of demand response program and which is suitable for the user?
- What are the demand response standards and how to choose the standard depending on one's need?
- What is the essential information required from the standard and how this information could be aligned with the facility infrastructure?
- How this information is sequenced between the demand response participants and which communication model is used to transmit the information?
- How the facility visualize their demand response program?

1.3 Objectives

Having the available standards for demand response program and the communication technology, the bottleneck is how these standards can be integrated to an existing manufacturing facility by providing demand response as a service to the utilities? and how web technologies would be beneficial in integrating the supply and the demand side?

Therefore, the objective of the thesis would be explicating a generic methodology for demand response communication infrastructure model between the demand response participants.

1.4 Assumption and Limitation

The demand response program involves actors such as Energy Market (EM), Distributed System Operator (DSO), Transmission System Operator (TSO), Demand Response Aggregator (DRA), Utility Service Provider (USP), and facilities. This thesis work focuses on the communication infrastructure between a single USP and a facility. A conceptual utility side is considered. The Utility Service Provider participating in demand response provides the cost of electricity a day ahead (Time based DR program) to the facilities depending on the energy market price. The electricity market data from [8], [9] are referenced for the electricity price signal. In facility, a system operator, responsible for Manufacturing Execution System (MES) assumed to have knowledge of production scheduling. Furthermore, the data and the communication model are developed based on the Fastory resource in Tampere University of Technology.

1.5 Thesis Outline

This thesis is structured as follows. Chapter 2 gives a literature review about demand response program. Different demand response standards and communication technologies will be studied in detail. In Chapter 3, the methodology for constructing the communication infrastructure for demand response will be described. The approach, and tools and frameworks will also be discussed in this same chapter. In the following Chapter 4, the implementation of the designed infrastructure will be discussed. Chapter 5, discusses the result from the communication model. Finally, Chapter 6 concludes with the future needs and scope of demand response.

2. LITERATURE AND TECHNOLOGY REVIEW

This chapter focuses on the literature and technologies behind the methodology and implementation of Demand Response program this thesis. Firstly, we study the demand response program and classification of DR program. Then electricity market structure and DR participation is presented with state of art. Later standards and communication protocol of DR program is studied with state of art approach. Finally, web-based portlet application is discussed to exhibit the DR program to end user.

2.1 Demand Response

According to Federal Energy Regulatory Commission, Demand Response (DR) is defined as [10]:

"*Changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized*."

According to U.S Department of Energy demand response is defined as [11]:

"a customer's opportunity either to reduce or shift their electricity usage during peak periods in response to time-based rates or other forms of financial incentives".

2.1.1 Demand Response Classification

Based on how the residential and industrial facilities change their electric usage, Department of Energy (DOE) has divided the DR program as show in the Figure 1:

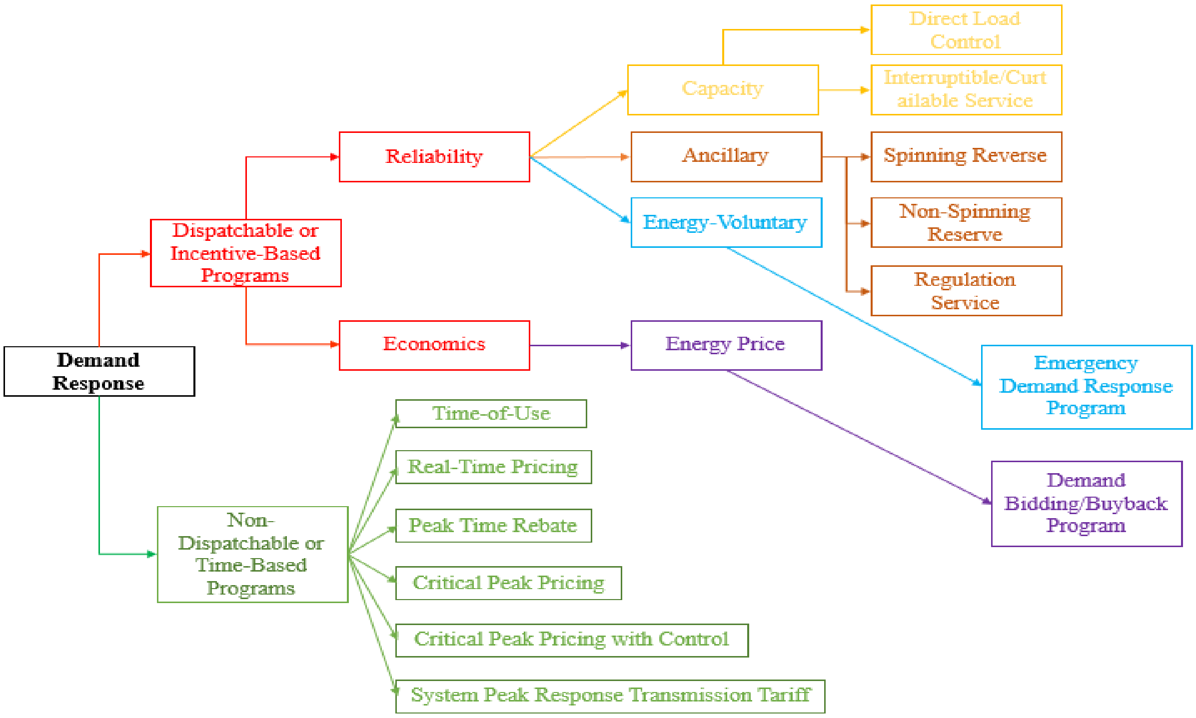


Figure 1. *Classification of DR program [10], [12], [13]*

2.1.1.1 Incentive-based DR program

From Figure 1, the Electricity Consumers (EC's) participating in dispatchable or incentive demand response program allows the facility operators to monitor and control their consumption pattern. For controlling the facilities electric devices, either Direct Load Control (DLC) program or Interruptible/Curtailable (I/C) service are used.

A participation payment is provided by the system operators to the consumers who are enrolled in DLC program. The consumer may override the DLC program with reduced incentive payment. [14] demonstrates a real-time DLC incentive-based DR program with home energy management system. Three simulated scenarios and EC's financial benefits were compared. The DLC approach is less considered by the EC's because of their electricity consumption privacy.

The EC's pay penalties if they override during I/C service and Capacity Market Programs (CMP) are provided to EC's when they curtail their load when grid contingencies arise [13]. An economic model was proposed in [15] for I/C and CMP program. The ISO uses the result of the model to study the behavior of the EC's corresponding to the incentives and penalties provided by the electricity suppliers.

Ancillary DR service allows the EC's as operating reserves. The consumers load curtailment bid are placed in ISO/RTO spot market. If the ISO/RTO accepts the consumers and their bids, the electricity market value is paid to them on participation in the DR program. [16] explains the electricity market policies and barriers to DR ancillary service.

During the reliability-triggered events the Emergency DR program (EDRP) are activated. An innovative Analytic Hierarchy Process (AHP) is proposed by the author in [17] to design EDRP. Incentives are provided to the EC's for measured load curtailment during those events. Unlike I/C service the EC's may or may not levy penalties if they override EDRP.

The Demand Bidding/Buyback (DB) program allows the consumers to bid directly in the wholesale electricity markets [13]. The impact between DB program and Market Clearing Price (MCP) is analyzed in [18]. The analysis from the model shows EC's profit, reduction in MCP and electricity generation cost.

2.1.1.2 Time-based DR program

From Figure 1, the Non-dispatchable or Price-Based demand response programs (PBP), the price of electricity fluctuates depending on the electricity production. In general, the Time of Use (TOU), Critical Peak Pricing (CPP), and Real-Time Pricing (RTP) are PBP's dynamic price programs. A higher electricity price is offered to EC's to reduce the demand during the on-peak period and lower price is offered during off-peak periods.

In Time-of-use DR program the 24-hour day is divided into different block of times. The rates are pre-determined months or years ahead which reflects the average electricity production and distributing cost [13]. Modeling DR program by TOU program is described in [19]. The [20] and [21] studies carried on residential EC's shows the ineffectiveness of TOU demand response program.

The utility at the earliest provides pre-specified electricity rates to EC's. The pre-specified duration will have the CPP which are very high. However, the EC's receives discount during non-CPP periods [13]. CPP demand response is not widely used as a DR strategy because of the fluctuation in the daily electricity market.

The EC's receives price of electricity a day-ahead in RTP or Hourly Price (HP) program [13]. The results from [22] argues that Hourly Price DR program engages the EC's more efficiently in Demand Side Management.

2.1.2 Demand Response Benefits

The following Table 1 describes DR benefits. From [13] and [23] the efficiency achieved by DR program is categorized into three types:

Types of Benefits	Recipient	Description
Financial benefits	EC's participating in DR program	(1) Electricity bill saving. (2) Incentive payment from utilities. (3) Indirect financial benefits by reducing congestion in the electric grid.

	Electricity Market	(1) Because of efficient use of resource and electric system, the electricity supply cost is reduced. (2) Indirectly reduces in building additional production, transmission and distribution lines.
Reliability benefit	Some or all EC's	The forced outages are reduced for all the electricity consumers and the grid reliability is maintained by different energy sources.
Environmental and Other benefits	Some or all EC's ISO/RTO	(1) More innovation in retail energy market. (2) Electric consumption reduction and emission deduction in high-polluting manufacturing facilities in peak-hours

Table 1. *Demand Response program benefits*

2.1.3 Demand Response in Wholesale Electricity market

It is essential to have the knowledge of wholesale electricity market structure to understand Energy market DR program (EMDR). Figure 2 shows the participants in wholesale electricity market. The generation, transmission and distribution companies can be grouped together as energy suppliers. The wholesale electricity market exists when the energy suppliers compete among themselves while offering electricity to the retailers. The retailers who are Independent System Operator (ISO) or Regional Transmission Organization (RTO) re-structure the electricity market price when selling to the consumers.

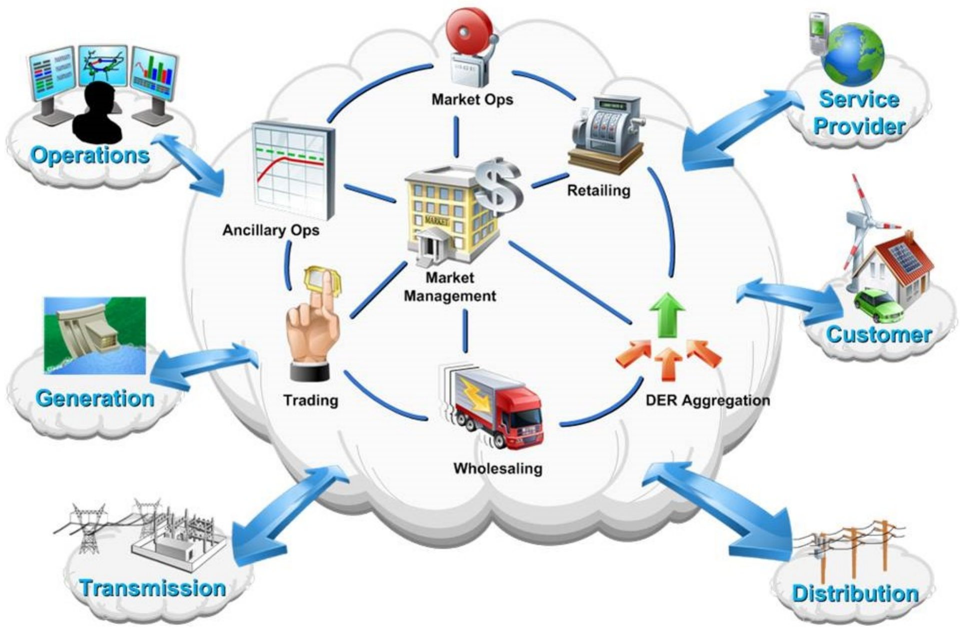


Figure 2. *Electricity Market Structure [24]*

The Distributed Energy Resource (DER) Aggregators are group of companies who buys electricity from the retailers at a negotiable price.

Different types of wholesale electricity market (Ancillary Service Market, Real-Time Imbalance Market, Transmission, Hour Ahead Market and Day Ahead Market) is discussed in [24]. However, as mentioned in Chapter 1.4 only the Day Ahead Market is considered in constructing the communication infrastructure for DR program in this thesis.

2.1.4 Day Ahead Market

As shown in Figure 3, Day Ahead Market is a category of electricity spot market.

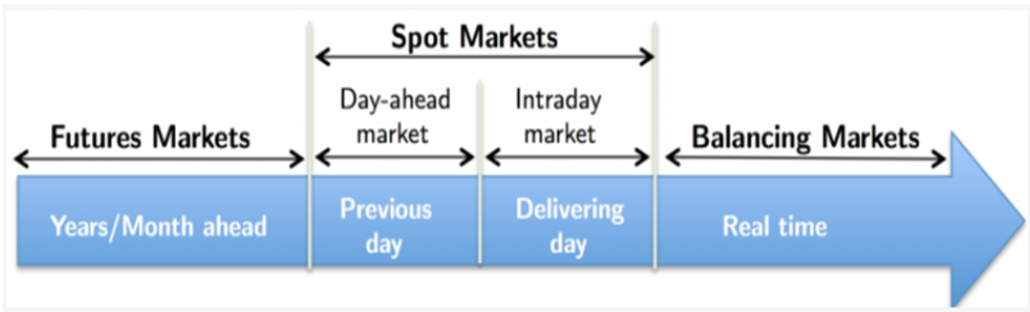


Figure 3. *Types of spot market in wholesale electricity market [25]*

In Day ahead market, the retailers buy and sell the electricity from the energy suppliers to the EC's a day ahead or 24 hours ahead [24]. The 24 hour is divided into hour by hour and each one hour have different electricity price.

As mentioned in [26] the hourly price of electricity for the next day is announced by the utilities after noon or later. Figure 4 gives an overview of distributed electricity price over day ahead hours.

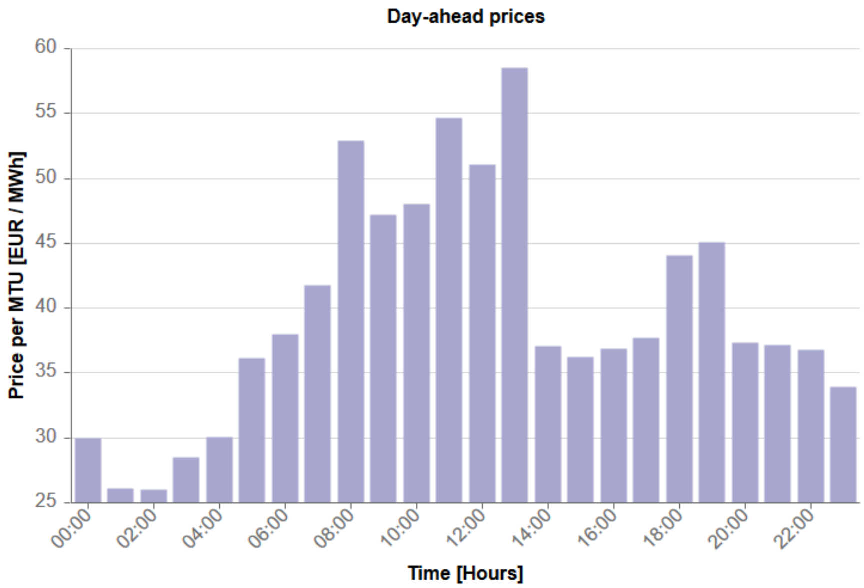


Figure 4. *Day ahead electricity price*

2.1.5 DR program Participation

As shown in Figure5, the EC's can engage in the Demand Response program by one of the following ways [27],

Consumers Provisioned Model – The EC's directly participate in the DR program with their on-premises available infrastructure.

Traditional Utility Model – The EC's sign-up to with the utilities. The utilities provide the infrastructure and the day ahead market price to the EC's to participate in the DR program.

Third-Party Aggregators – Two or more party specialized in DR participation aggregate and form third-party Demand Side Response Aggregator (DSRA). The DSRA make their own policies with the EC's and provides the DR program to System Operators.

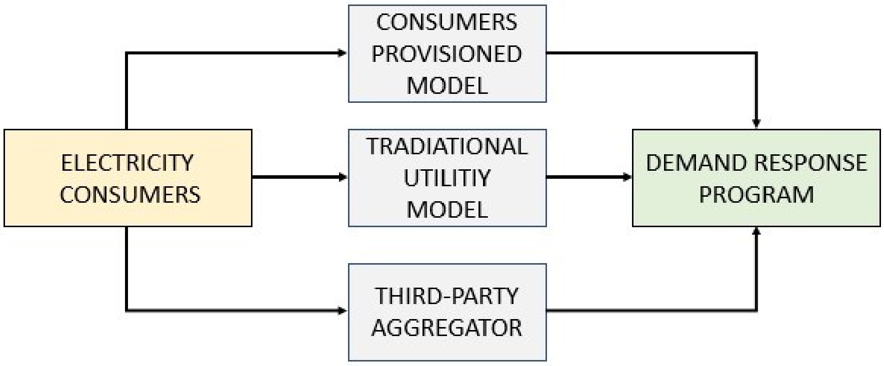


Figure 5. *EC's participation in DR program*

Based on the business model [28] and information model the DR program participants are shown in Figure 6.

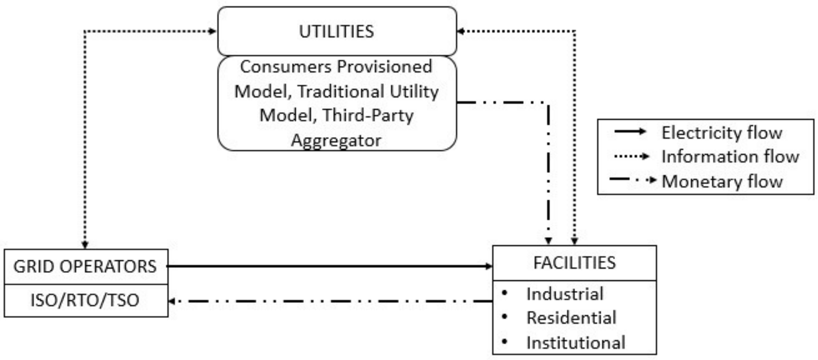


Figure 6. *DR Participants*

The Grid Operators are electricity suppliers and they decide the wholesale electricity market. The Utilities are composed of the DR service providers who monitors the EC's electricity usage. They alert the EC's when the grid is unstable or during the high electric price depending the EC's demand response enrollment. The Facilities are actual Electricity Consumers who gets incentives and participate in the DR program.

2.1.6 State of Art Demand Response Participant Modeling

The article [29] examines a Simulated Demand Response of a Residential Energy Management System (REMS). The proposed system is a discrete event system that can able to monitor, plan and control the energy consumption in residential buildings. The authors in the article proposed "components" to be used for the REMS system. The components include actors and equipment. Actors were divided into primary (Owner, utility and REMS

Controller) and secondary (dishwasher, plug-in-electric vehicle, water heater, and household equipment). According to the authors, the primary actors initiate the system events and the secondary actors respond to those events. A user interface was designed to the end user to monitor the energy consumption and to override the equipment status. The authors analyzed the residential appliance electricity demand and their operating duration. There are mainly two functions provided by the REMS system. First, the system delays the equipment that was not utilized at the time of Demand Response service time. Second, during the Demand Response service time, energy dissipates from the stationary battery to the equipment. The authors used four scenarios to define the REMS system. The first scenario was constructed with the normal electricity consumption without any programmable thermostat. The next scenario integrates the programmable thermostat to the normal electricity consumption. The third was constructed with the REMS and the equipment's, and the final scenario with the REMS and a stationary battery. The authors conclude from the simulated result that, the REMS can provide an energy consumption reduction during the DR Service time. Furthermore, cost-benefit analysis was carried out by the authors that showed payback period for the consumers by participating in the Demand Response program.

The article [30], proposes a model of Demand Response (DR) Energy Management System in the Industrial Facilities. The effects and the importance of Demand Response in Industrial sectors were discussed by the authors. To have a common understanding of DR program, a model element and a model architecture were defined by the authors. Initially each model elements were identified by unique graphical symbols. Later the authors labelled each model element. Finally, they constructed an architectural overview representing each model elements and their interrelationship. The authors modelled the production flow as a State-Task Network (STN) that defines two types of nodes. A "state node" that represents the inputs, intermediate products, and the final products; and a task node that represents the manufacturing process. The authors further classified the task nodes into Scheduling task (process that can be scheduled prior with respect to electricity demand) and Non-Scheduling task (process that cannot be scheduled prior with respect to electricity demand). The authors finally described the developed model with the help of a steel manufacturing facilities.

A series of DR messages exchanged between the utility and facility side corresponding to dynamic real-time electricity price was provided by the authors.

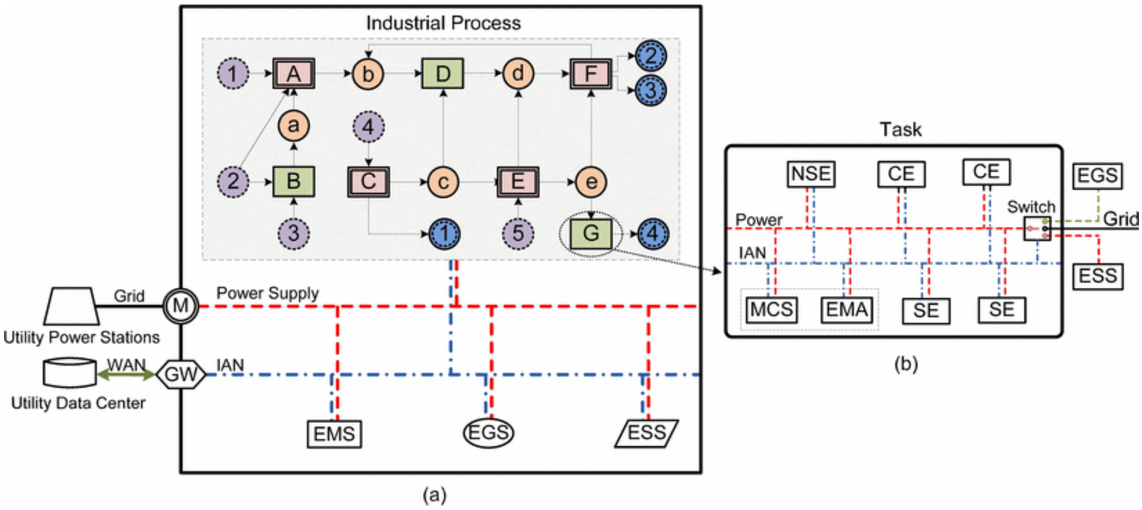


Figure 7. *Model of DR program in industrial facility [30]*

Using the model elements and the model architecture, the authors were able to construct the pictorial representation of the demand response program in the steel manufacturing facility.

The authors [31] in proposes production planning model with Time Of Use DR program in air separation plant and cement plant industrial facilities. The authors of [32] discussed various smart grid technologies and presented case studies of ancillary service and food processing industrial facility participating in DR program.

The difference between old and new electricity market is discussed in reference [33]. [6] and [36]outlines the problem in implementing DR program and emphasis the need of DR architecture. Furthermore references [35]–[38] describes more on DR concepts and models. Even though the references explain the modeling of DR participants, the interaction between the participants were not studied.

2.2 DR Standards and Communication Protocol

Demand Response standards defines the Information and Communication Technology (ICT) required between the Utilities and Facilities participating in the DR program. The following sub-section describes the available DR standards and mapping between them. These standards were derived to deliver interoperable, transparent, consistent and reliable DR service between the DR participants.

2.2.1 OASIS Energy Interoperation

The Organization for the Advancement of Structured Information Standards (OASIS) formed the Energy Interoperation Technical Committee [39], to define the information and communication models between all the actors involving electricity as shown below in Figure 8.

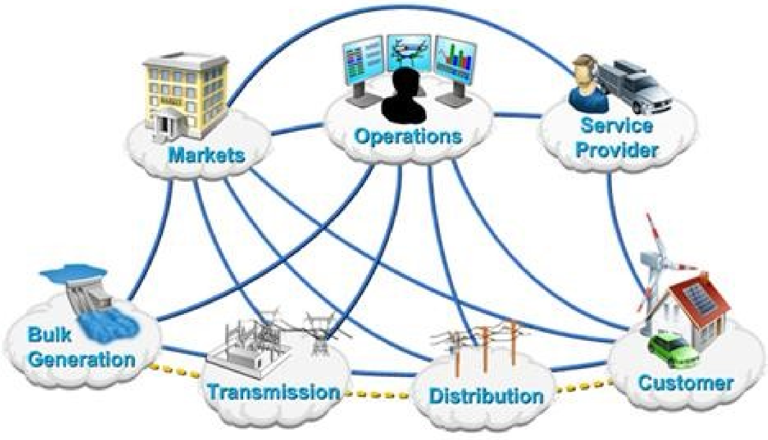


Figure 8. *Actors involving in OASIS Energy Interoperation*

The message corresponding to day ahead market electricity price, time of use and DR events communicated between the actors are derived from Energy Market Information Exchange (EMIX) specification and Web Service Calendar (WS-Calendar) standards. These standards are incorporated with the OASIS Energy Interoperation standard (OASIS EIOp). The transmitting and receiving DR service message uses Extensible Markup Language (XML) schema [40] defined by World Wide Web Consortium (W3C).

2.2.1.1 WS – Calendar

WS – Calendar defines the DR program schedule (when to participate in DR program?) and duration (how long to participate in DR event?) WS – Calendar specification is used in EIOp. More information about OASIS WS-Calendar Technical Committee (TC) can be found in [41]. The terms and description of WS – Calendar with EIOp context is described in following Table 2.

Term	Description
Component	Information structure that have Component and Parameters.
Duration	Length of DR service offered to EC's
Interval	A single discrete segment in DR service represented by duration
Sequence	A set of intervals having links and relationship which can be relocated
Gluon	The serialization of intervals in a sequence is influenced by Gluon
Availability	A parameter representing when an EC's can schedule their resources to participating in the DR service.

Table 2. *WS – Calendar with EIOp context [39]*

The simplest way to express the DR service in a constant pattern over constant interval in a sequence with WS – Calendar is shown below:

Start:	10:00	Duration:	1 Hour
		Duration:	1 Hour
		Duration:	1 Hour

Table 3. *WS – Calendar DR sequence of 3 intervals*

2.2.1.2 EMIX

EMIX defines the schema of power and energy market for DR program. The EMIX 1.0 Specification in OASIS EIOp consists of four schemas:

1. **EMIX schema** – describing electricity market context.

2. **SI Scale schema** – describing a measurement scale given by the System International (SI).

3. **Power schema** – describing the information to be exchanged based on the EMIX framework used in electricity market between the electricity suppliers and ISO/TSO's. The

4. **Resource schema** - describing specific resources that affect energy market.

More information on EMIX 1.0 can be found in [42]. The simplest way to express the power information is shown below.

Units	KW	Quantity	30

Table 4. *EMIX DR representation*

When integrating the WS – Calendar and EMIX standards to represent the basic market and power information in the DR service can be represented as in Table 5:

Units	KW	Start:	06:00	Duration:	1Hour	Quantity:	40
				Duration:	1Hour	Quantity:	80
				Duration:	1Hour	Quantity:	30

Table 5. *Simple WS – Calendar and EMIX in DR program*

The above Table 5 show the unit of power KW starting form 06.00 with the duration of one-hour in one- hour interval of varying power quantity.

The EMIX message schema is detailed in APPENDIX A – EMIX Classes and Message SchemA

2.2.1.3 OASIS EIOp Actors

All the interactions of DR program between the DR participants are involved between Actors is a pairwise interaction. The Actors are classified as **Virtual Top Node (VTN)** and **Virtual End Node (VEN).** In any DR interactions, there will be one VTN and remaining actors in the DR interaction is considered as VEN.

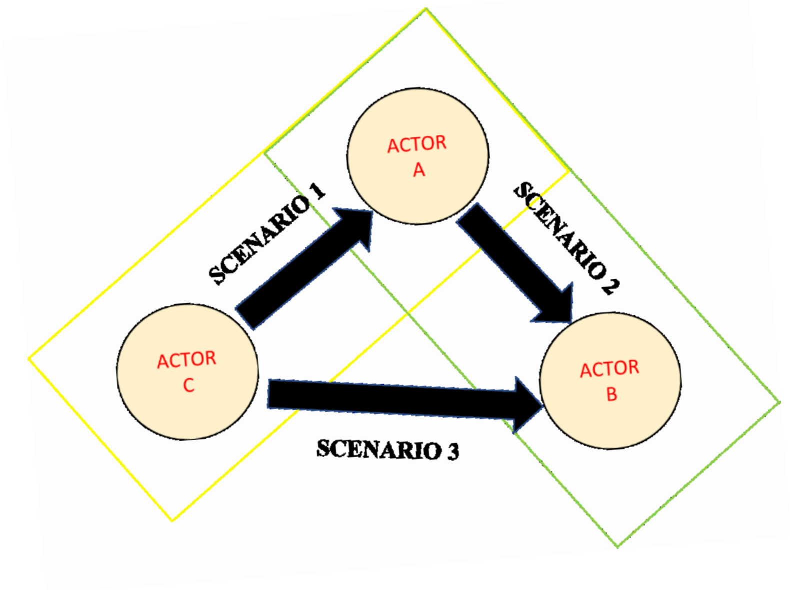


Figure 9. *DR program interaction*

The above Figure 9 shows three different scenarios of DR program actors pairwise interaction. In scenario 1, Actor C is the Virtual Top Node with respect to Actor A which is Virtual End Node. Here Actor B is not interacting in the DR program. Likewise, in scenario 2, Actor A is VTN with respect to Actor B and in scenario 3, Actor C is VTN with respect to Actor B. However, in a DR program many actors involve, and Figure 10 below shows the complex interaction pattern between the DR participants.

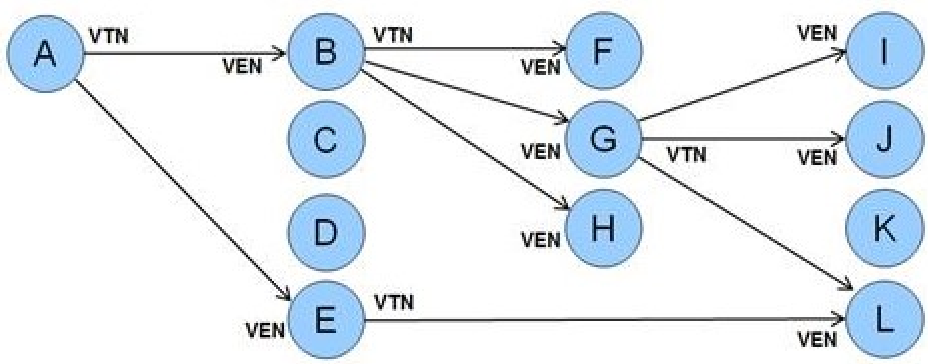


Figure 10. *Complex DR program interaction [39]*

The role for each actor from the above figure could be as given in the following table,

ACTOR	ROLE	DESCRIPTION
A	Independent System Operator	Top level actor who is providing the market price data for DR program.
B, C, D, E (VEN)	Facilities	The EC's participating in DR program
B, E (VTN)	Utility/ Aggregator	Utilities or aggregators who initiate the DR event to EC's
F, G, H (VEN)	Facilities	The EC's participating in DR program. Could be Industries, Commercial buildings or Residential EC's.

G (VTN)	Controllers, Switches, PLC's	Controllable device for the resources in the facilities.
I, J, K, L (VEN)	Resource	The actual devices that curtail the electricity when DR event is called upon.

Table 6. *DR actors and their roles [39]*

2.2.1.4 VEN Load Response in DR program

When the DR event for Day ahead market price is called the, VEN response will be as shown in Figure 11.

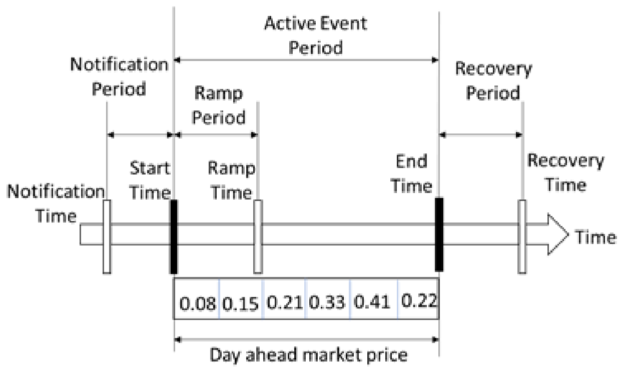


Figure 11. *VEN's Load Response [39]*

The load response period is described in the following Table 7.

DR program periods	Description
Notification Period	Duration when a VTN notifies the VEN to participate in the DR program.
Ramp Period	Duration when the VEN's starts to shift their normal electricity usage to its load curtailment state. If negative ramp period is assigned, the load curtailment shifts directly to Active Period.
Active Period	Duration between the actual DR event Start time and End time load curtailment process.
Recovery Period	Duration when the VEN's returns to its baseline consumption.

Table 7. *DR load response period*

2.2.1.5 OASIS EIOp Services

As mentioned above, actors involved in the DR program is responsible for transmitting and receiving DR information between them. The information of demand response program is divided into various DR services. The services are invoked and provided to consumer either by VTN's or by VEN's. And for any service there exist different request and its associated response operations.

Major OASIS EIOp services are as follows:

- **Transactive Services:** Consist of EiRegisteration services, EiTender and EiQuote Services and EiTransaction Services, and its associated operations.
- **Enrollment Services:** Consist of EiEnroll Service and associated operations.

- **Event Services:** Consist of EiEvent service and its associated operations.
- **Report Services:** EiReport services, EiHistorial Service and EiProjection Services, and its associated operations.
- **Event Support Services:** EiAvailablity Service and EiOpt Service, and its associated operations.
- **Market Information Services:** EiMarket Context Services and its associated operations.

The following table shows the different demand response operations associated with the above-mentioned services.

Service/ Description	**Operation**	**Response**	**Service Consumer**	**Service Provider**
EiRegisteration: To identify the DR participants. The operation involves participants to request, create or cancel DR participants registration.	EiCreateParty Registration	EiCreatedParty Registration	Party[2]	Party
	EiRequestParty Registration	EiReplyParty Registration	Party	Party
	EiCancelParty Registration	EiCanceledPartyRegistration	Party	Party
EiTender: An invitation to parties to participate in DR program that leads to transaction	EiCreateTender	EiCreatedTender	Party	Party
	EiRequestTender	EiReplyTender	Party	Party
	EiCancelTender	EiCanceledTender	Party	Party
	EiDistributeTender	----------------	Party	Party
EiQuote: The day ahead market price signal sends to parties.	EiCreateQuote	EiCreatedQuote	Party	Party
	EiRequestQuote	EiReplyQuote	Party	Party
	EiCancelQuote	EiCanceledQuote	Party	Party
	EiDistributeQuote	-----------------	Party	EiTarget
EiTransaction: The operation to manage transaction in the transactive service	EiCreateTransaction	EiCreatedTransaction	Party	Party
	EiRequestTransaction	EiReplyTransaction	Party	Party
EiEnrollment: Operations that happens after registration of the parties to establish demand response interaction.	EiCreateEnroll	EiCreatedEnroll	Party	Party
	EiRequestEnroll	EiReplyEnroll	Party	Party
	EiCancelEnroll	EiCanceledEnroll	Party	Party
EiEvent:	EiCreateEvent	EiCreatedEvent	VTN	VEN
	EiChangeEvent	EiChangedEvent	VTN	VEN
	EiRequestEvent	EiReplyEvent	VTN/VEN	VTN/VEN

[2] "Party" can either be a VTN or a VEN involving in demand response program interaction.

The core information payload that communicates between the demand response program participants to create, change, cancel, request DR event.	EiRequestPending Event	EiReplyPending Event	VTN/VEN	VTN/VEN
	EiCancelEvent	EiCanceledEvent	VTN	VEN
	EiDistributeEvent	---------------------	VTN	VEN
EiReport: Report generated between the participants in DR program either dependent or independent of any EiEvent that can be requested and response any time during the interaction.	EiCreateReport	eiCreatedReport	VTN/VEN	VTN/VEN
	EiUpdateReport	EiUpdatedReport	VTN/VEN	VTN/VEN
	EiRequestReport	EiReplyReport	VTN/VEN	VTN/VEN
	EiCancelReport	EiCanceledReport	VTN/VEN	VTN/VEN
	EiCreateProjection	EiCreatedProjection	VTN/VEN	VTN/VEN
	EiCreateHistorian	EiCreatedHistorian	VTN/VEN	VTN/VEN
	EiRequestHistorian	EiReplyHistorian	VTN/VEN	VTN/VEN
	EiCancelHistorian	EiCanceledHistorian	VTN/VEN	VTN/VEN
EiAvailaiblity: Show whether VEN is available to execute the called DR program. The EiAvailablity service operation could ACCEPT, REJECT or RESTRICT the EiEvent.	EiCreateAvail	EiCreatedAvail	VEN	VTN
	EiRequestAvail	EiReplyAvail	VEN	VTN
	EiCancelAvail	EiCanceledAvail	VEN	VTN
EiOpt: All the EiAvailability resources participating in DR program must have one OptIn and OptOut options. The VEN's can Optin or OptOut during the ongoing the DR EiEvent.	EiCreateOpt	EiCreatedOpt	VEN	VTN
	EiRequestOpt	EiReplyOpt	VEN	VTN
	EiCancelOpt	EiCanceledOpt	VEN	VTN
EiMarketContext Service: Provides the whole information regarding the electricity	EiRequest MarketContext	EiReply MarketContext	VEN/VTN	VEN/VTN

market as given in EMIX standard.				

Table 8. OASIS EIOp service description[39]

A generic interaction of any service mentioned above between party or VEN and VTN is show in the following Figure 12.

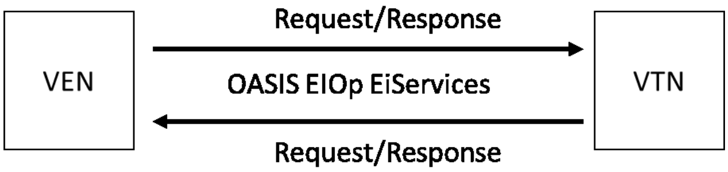


Figure 12. Interaction between actors in DR program

2.2.2 OpenADR

Open Automated Demand Response (OpenADR), developed by Demand Response Research Center (DRRC) in Lawrence Berkeley National Laboratory, is a standard communication data model used by the DR program participants for sending and receiving the DR signals automatically. However, the standard does not expect a fully automated system on the facilities. Previously, OpenADR 1.0 framework was created which lacked interoperability between the DR communication devices. Later OpenADR 2.0 standard was developed to acquiescent National Institute of Standards and Technology (NIST) Smart Grid framework. The OpenADR 2.0 extended to Profile a and Profile b. While the OpenADR 2.0a supports low embedded devices with constrained DR service, the OpenADR 2.0b supports high embedded devices with all DR service including reporting service. Therefore, OpenADR 2.0b profile is used widely as DR communication protocol. In the profile, day-ahead demand response program is referred as Slow DR. While calling DR event by a utility to facility, if the response lead time in seconds, then it is referred as Fast DR. The OpenADR standard is a subsection of OASIS EIOp which includes EMIX and WS-Calendar specifications [43] [44]. This can be represented as in Figure 13.

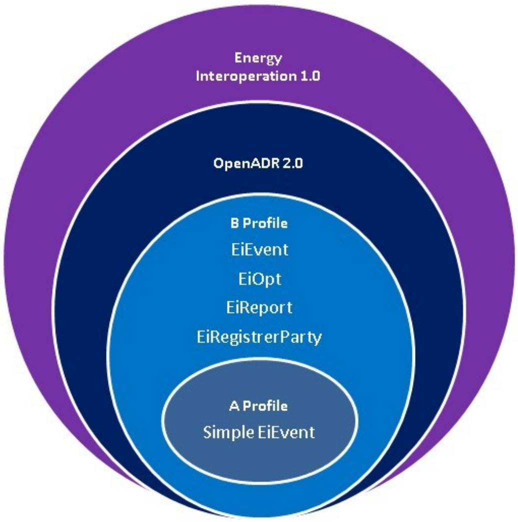


Figure 13. *Subset representation of OASIS EIOp and OpenADR [45]*

The OpenADR demand response program exchanges DR signals in human and machine readable eXtensible Mark-Up Language (XML) [40]. It follows Client-Server architecture, meaning VEN's are the clients and VTN's are the servers. The client/service architecture communicates in PUSH/PULL pattern between the DR participants over Hyper Text Transfer Protocol (HTTP) or XML Messaging and Presence Protocol (XMPP) transport [46]. When the DR signal are transmitted to the clients (VEN), the resources in the client side (facility) switches to a pre-programmed state. In pre-programmed state, the resources in the facility starts to curtail their load below their baseline consumption.

Figure 14 shows the OpenADR communication architecture. The two-way interaction between the VEN (Site A, B, C, D, E) and VTN (ISO or Utility, aggregated loads) is through standardized Application Program Interface (API). The Demand Response Automation Server (DRAS) is the logical OpenADR server that provide DR service between the DR program participants.

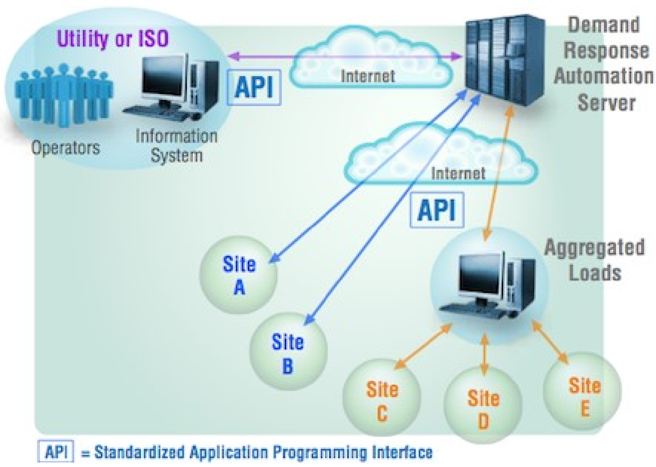


Figure 14. *OpenADR Communication Architecture [47]*

2.2.3 SEP2.0

IEEE 2030.5™-2013 (Smart Energy Profile 2.0) defines the communication standard between the smart grid and the EC's [48]. The information communicated between them are the EC's electricity usage, efficiency, DR program information and price signals. The SEP 2.0 enables the smart appliance in EC's facilities to integrate with the DR applications. The profile is independent of physical devices utilizing the Transmission Control Protocol/Internet Protocol (TCP/IP) as their transportation layer (will be discussed later in this chapter).

SEP profile are divided into independent function sets. The profile follows Client-Service architecture and for a function set any device could be server and/or a client [49]. Some of the function sets are price communication, Demand Response and Load Control, Energy Usage Information, Service Provider Messaging. More of these function sets can be found in [48].

2.2.4 BACnet

Building Automation and Control network is a communication protocol by ISO 16484/ ASHRAE 135 (American Society of Heating, Refrigerating and Air-Conditioning Engineers) [50]. The BACnet protocol allows different equipment manufacturers to communicate in Building Automation and Control Systems (BACS). The equipment's application could be HVAC (Heating, Ventilating, Air-Conditioning) control, lighting control, fire detection, smart elevators, and/or access control.

In BACnet [51] "objects" are defined as physical sensor inputs, outputs and software process, and each "objects" have set of "properties". The property of any object defines the status an equipment and the status are communicated to other the devices via BACnet protocol. There are 60 objects in BACnet specification and DR program uses Load Control Object (LCO) for building load curtailment applications. Table 9 gives an overview of Load Control Object Types with Conformance code indicating the object property should be R – Readable or W – Readable and Writable or O – Optional, that can be integrated to DR program. More properties can be found in [52].

Property Identifier	Conformance Code
Object_Identifier	R
Object_Name	R
Object_Type	R
Description	O
Present_Value	R
State_Description	O

Status_Flags	R
Event_State	R
Start_Time	W
Shed_Duration	W
Duty_Window	W

Table 9. *BACnet Load Control Object property [52]*

2.2.5 Relationship between DR standards

Article [53] extends the IEC 61968 Common Information Model (CIM)/ Distribution Management standard to inherit the Metering Package to Metering Asset Class of OpenADR 2.0b profile as shown in the Figure 15.

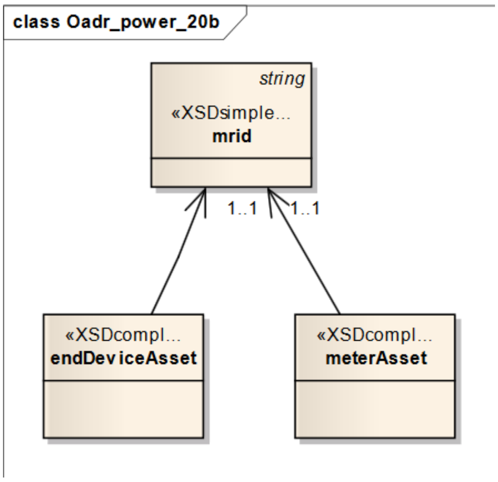


Figure 15. *IEC 61968 mapping to OpenADR*

The higher-level information mapping between OpenADR and OASIS EIOp is shown in Figure 16.

OpenADR	OASIS Energy Interoperation
Participant	EI Party
DRAS Client	EI Registration
DR Event	EI Event
Opt Out State	EI Opt Out
DR Event Feedback	EI Feedback
	EI Quote
Bid	EI Tender
	EI Contract
Program	EI Program
Program Constraints	EI Constraint
	EI Usage

Figure 16. Mapping between OpenADR and OASIS Energy Interop [54]

2.2.6 OSI Model

Open System Interconnection (OSI) Model is a conceptual communication model used to identify and separate any two-party involved in communication irrespective of the technology used. The model is separated into seven layers. First three layers (Application, Presentation and Session layers) are called as Upper Layers and next four layers (Transport, Network, Data link and Physical layers) are called as Lower Layers. On receiving transmitted data, each layer does a specific operation those data and transfers to the next layer.

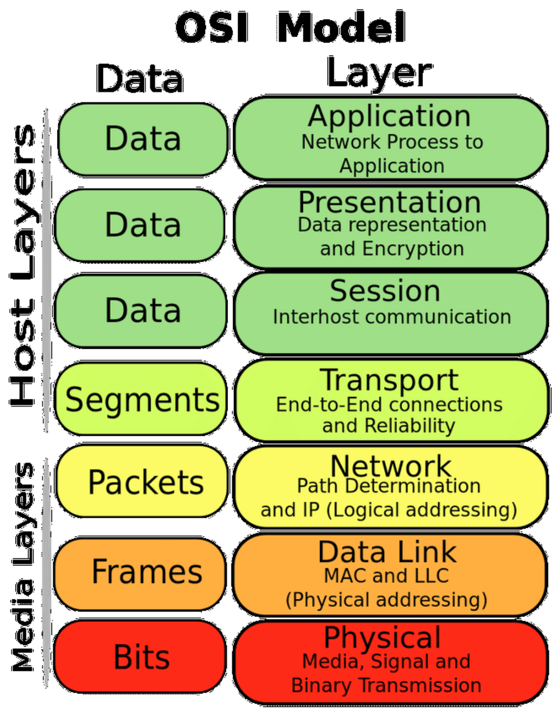


Figure 17. OSI Model [55]

Application layer provides HTTP, FTP service in the communication network. Presentation layer gives a representation of the transferring data in the network like JPEG, ASCII formats. The Session layer is responsible for opening and closing the network connection between the applications. REST (Representational State Transfer), Web Socket, and RPC (Remote Procedure Calls) are some used in session layer. Transport layer provides an end to end communication between the end device via network. TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are mostly used transfer layer protocols. IP (Internet Protocol) version4, IPv6 are in Network layer providing a logical address for the device in the network. MAC (Media Access Control) and LLC (Logical Link Control) in Data Link layer provides physical and unique address to the device in the network communication. The bit data are transmitted and received between the hardware system and the network in the physical layer. Ethernet, Control Area Network and Universal Serial Bus are in physical layer.

2.2.7 Communication Protocol

To communicate the DR signals between the utilities, XMPP and HTTP application layer protocol are used [41], [49].

XMPP

XMPP stands for Extensible Messaging and Presence Protocol. XMPP is decentralized, secure, flexible, asynchronous and instant-messaging lightweight middleware protocol. XMPP uses Transmission Control Protocol and exchanges messages in XML format. A simple XMPP communication architecture is shown in Figure 18.

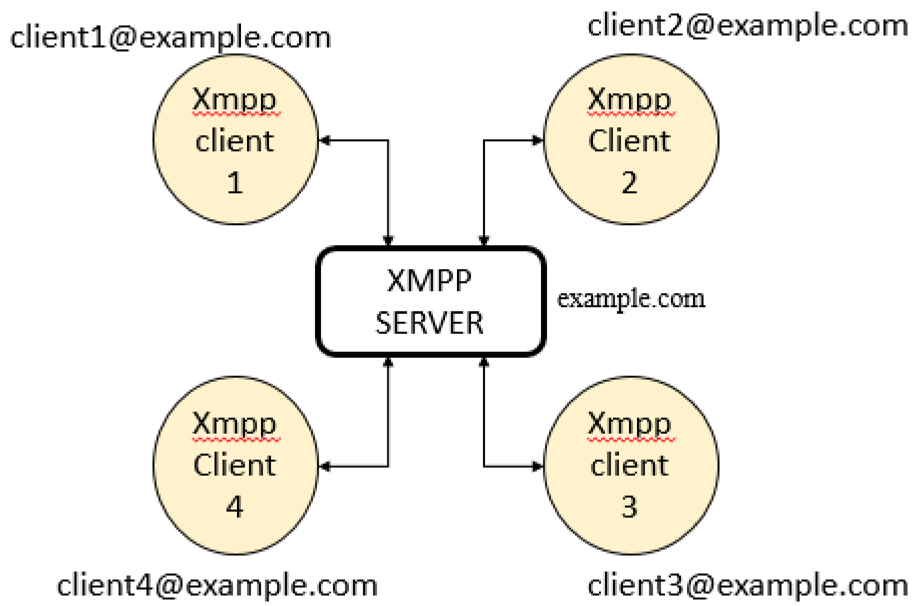


Figure 18. *XMPP Client-Server communication*

In XMPP, every client will have a unique name associated with its server. For example, as shown in Figure 18, client1, client2, client3, client4 is associated with example.com domain. The clients will have unique address after registering to the server's domain. A simple Client-Server communication message and its sequence is established in Figure 19.

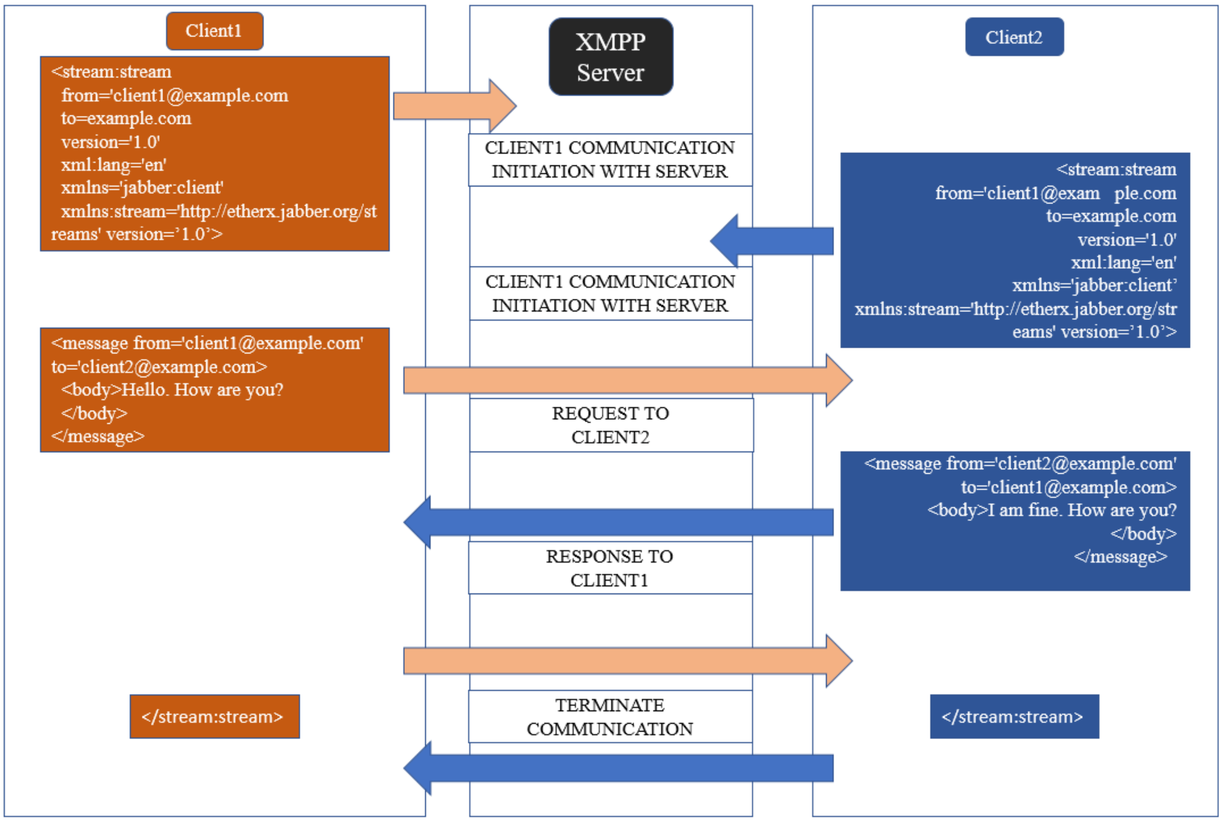


Figure 19. *XMPP Client-Server communication*

HTTP

HTTP - Hyper Text Transfer Protocol. HTTP is an application layer, request-response protocol over TCP/IP communication designed to communicate between client and server [56]. HTTP is asymmetric and stateless protocol. As defined by Request For Comments (RFC) -2616, HTTP,

"*an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers* [57]" .

To identify the communicating resources, URL (Uniform Resource Locator) is used. *protocol://url address or hostname: port/file path,* is the URL syntax. For example, http://www.example.com/index.jsp URL uses HTTP protocol with *www.example.com* as url address and port 80 (TCP default port number for HTTP) and *index.jsp* resource. GET, POST, PUT, HEAD, DELETE, PATCH and OPTIONS are HTTP request methods to communicate between client and server. While communicating, the client uses any of the request methods to server for the required service. A simple client-server HTTP communication sequence using GET method is shown in Figure 20.

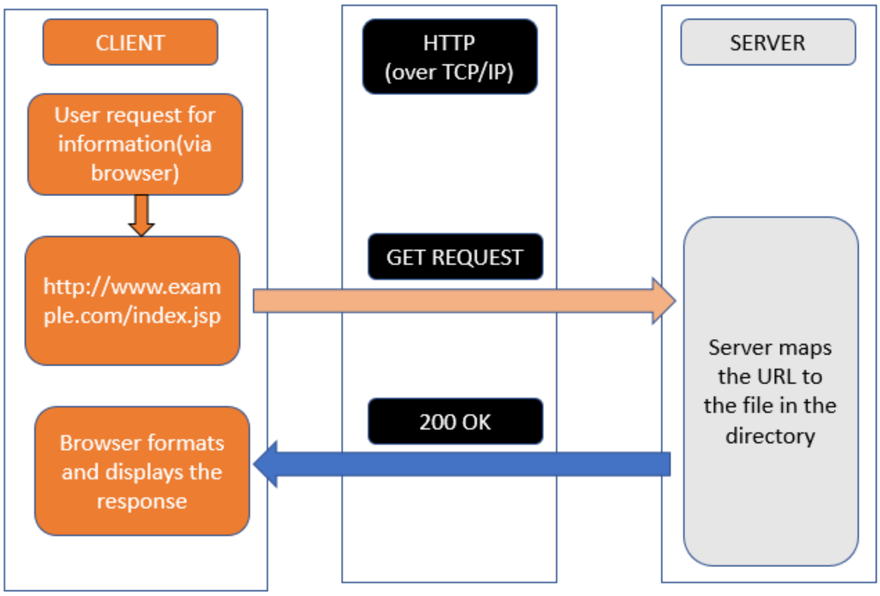


Figure 20. *HTTP client-server communication*

2.2.8 PUSH/PULL technology

PUSH-PULL communication model in Client-Server architecture is generally accomplished between consumer and producer by publish/subscribe - request/response of information. PUSH technology is referred as server push and PULL technology as client pull [58].

A simple PULL communication model can be referred from Figure 21 below,

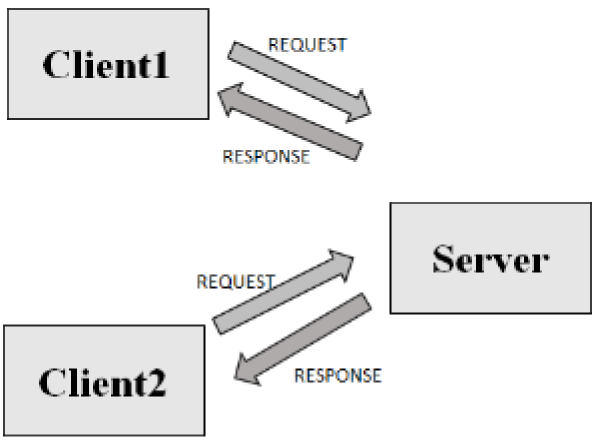


Figure 21. *PULL Communication model*

The client1 and client2 request (PULL's) for an information from the server. In response to the PULL request, the server sends the requested information to client1 and client2.

A simple PUSH communication model can be explained with the Figure 22 below,

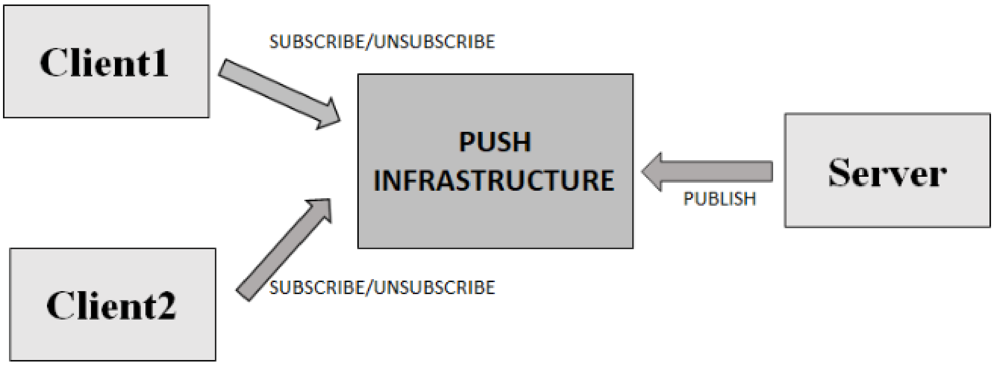


Figure 22. *PUSH Communication model*

The server publishes the available information into the push infrastructure. If the client 1 or client 2 are interested in any of the published information, they can subscribe from the PUSH infrastructure. The clients can also unsubscribe from the subscribed information if they want to opt out.

2.2.9 Portal and Portlet:

The portlets are pluggable user interface windows in the web-application portal that aggregate information from different source. The source for a portlet can either be unique to a single portlet or common to different portlets. All the portlets in a web-application runs inside a portlet container. The lifecycle of portlet in a portal container is shown in Figure 23.

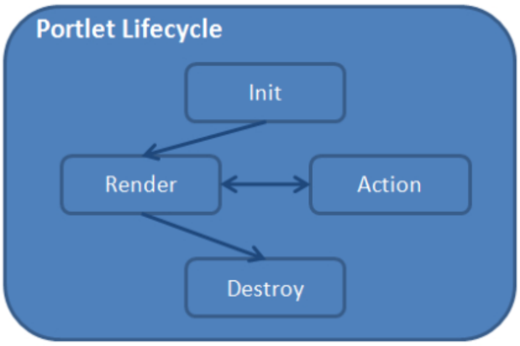


Figure 23. *Portlet lifecycle [59]*

Before any portlet is deployed in the portal, the container initiates the portlet by *init* method. The displaying of portlet content uses *render* method once a portlet is initiated.

The business logic of portlet is processed by the *action* method. The *destroy* method is called by the portlet container when a portlet is removed from the application portal. Comparison between portlet, servlet and widget was discussed in his thesis [60]. A web-application portal is shown in the following Figure 24 with different sized portlet which can aggregate different information in each portlet content.

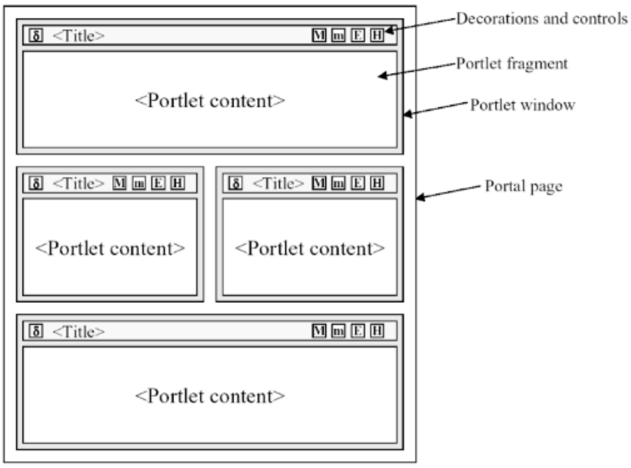


Figure 24. *Portlet structure*

2.2.10 State of Art DR Standards and Implementation

A demand response program for industrial devices could be modeled to shift the electricity consumption of some equipment during the high electricity price period without affecting the consumers. This successively adds value to the shop floor by saving cost on electricity usage by the resource present in the shop floor. The article [58] discuss two types of resource shift strategies namely Shift able Class First (SCF) and Controllable Device First (CDF). In SCF, the equipment's in the EC's facility with low priority are scheduled first in the load-curtailment process. After the process the agreed electricity reduction is compared with the actual reduction. If the agreed reduction is less than the actual reduction, the low priority equipment's curtail their loads until satisfying the actual reduction. In CDF, after all the loads shifted using SCF strategy, low priority controllable loads are scheduled to curtail their consumption. A flour mill equipment was experimented with these strategies. The result shows cost saving of approximately 25% in their annual electricity consumption before and after applying the control strategies in DR program.

A Multi-Agent System (MAS) architecture is presented in article [61]. OASIS EIOp standard is used as the communication language to exchange the information between the DR participants. They used XMPP as application layer protocol for instant messaging between the client and server. The client-server architecture was implemented using fifteen Raspberry PI boards. However, the proposed model has inadequacy when considering Quality of Service (QoS) and security.

The research conducted by Lawrence Berkeley National Laboratory on OpenADR suggest opportunities, barriers and actions needed by industrial EC's for DR program [62]. A multi-tier architecture was proposed in [63] with OpenADR 2.0 specifications. In the paper, basic two-tier architecture was demonstrated between the utilities and the end users with XMPP and HTTP communication protocols. The network traffic was analyzed with both the protocols.

The communication gateway for demand response program was developed in [64]. The paper explains the possibility to control the Building Automation and Control Systems (BACS) using BACnet Load Control Object. For utilizing the wireless communication architecture ZigBee protocol is used with BACnet. The author developed BACnet-ZigBee gateway to map the demand response application. The gateway provides interoperability between BACnet protocol and ZigBee wireless communication protocol. This gateway help to control the BACS. Finally, the demand response program was illustrated with a hardware and software architecture. The hardware consists of a power supply, ZigBee and BACnet module, I/O ports, Ethernet interface. The software architecture maps the BACnet objects to ZigBee event-frame through a translation routine.

Demand Response communication over Constrained Application Protocol (CoAP), an Internet of Things (IoT) standard protocol, is proposed in [65]. Constrained devices[3] participating in DR program can utilize this protocol for communicating demand response signals between them. The article presents a comparison between HTTP/XML and CoAP/JSON[4] protocol network overhead[5] while communicating OpenADR demand response signals. The EiReport demand response service is selected for communicating between VTN and VEN from 10 to 100 seconds to calculate the network overhead. The result from the comparison shows a 45% network overhead reduction using CoAP/JSON over HTTP/XML. CoAP standards are still evolving and it lacks security and reliability when compared with HTTP protocol. More research is needed on CoAP based DR communication.

Various DR standards, their interoperability, use case, consistency, communication protocol and business opportunities developed by IEC, IEEE and NIST are presented in [67] and [68]. They concluded by saying that the process for explicating the communication infrastructure by a unique standard is not possible. Therefore, a user group or alliance are synergized within these standards to develop their tailored communication infrastructure for their facilities DR program. The possibility of synergizing between OpenADR, Advanced Metering Infrastructure (AMI), OASIS EIOp is discussed in [47]. Earlier study

---

[3] Constrained Devices - Small devices with limited Central Processing Unit (CPU), memory, and power resources, are referred as "constrained devices". Sensors and Actuators can be constrained devices. These devices form a network, becoming "constrained nodes" in that network [66].
[4] JSON - JavaScript Object Notation, a Human-Machine readable data-exchange format.
[5] Network Overhead – Additional datagram needed to transfer the actual message between the communicating participants.

[69] shows the implementation of Demand Response Automation Server (DRSA) through integration between OpenADR 1.0 and BACnet.

2.2.11 Summary

With the available demand response standards, communication protocols and research researches conducted in demand response program, there are various ways to implement demand response program. However, the researches lacks in defining a generic way to construct a communication infrastructure. Next chapter provides a generic demand response infrastructure and an approach for the demand response participants to communicate with the utilities.

3. DEMAND RESPONSE INFRASTRUCTURE

Previous chapter discussed various demand response programs, demand response standards, and existing communication protocols. The scope of this thesis is limited to industrial facility as demand response participant and RTP as demand response program. The communication between participant facility and DR program utility should be robust, transparent, and user friendly. Therefore, a methodological approach should be followed to model the demand response program. The following section will discuss those approaches. Later, tools and frameworks for explicating the approach is discussed.

3.1 Demand Response Communication Infrastructure

The Figure 25 below depicts the overall demand response communication infrastructure. The infrastructure resembles OSI communication architecture to distinct the actors participating in demand response program.

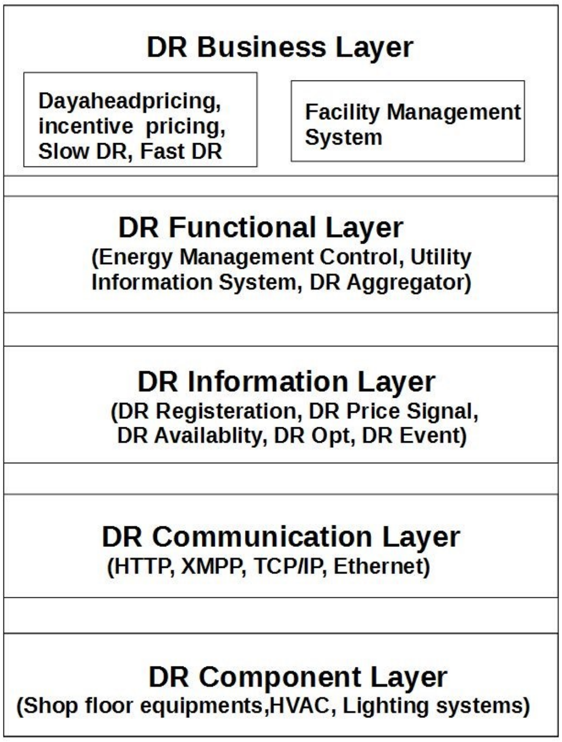


Figure 25. *Demand Response Communication Infrastructure*

The DR component layer represents all the resources in industrial facility that consumes electricity. This includes the shop floor equipment's (sensors, actuators, robots, machines, etc.), and HVAC and lighting devices. DR communication layer acts as a communication gateway between the component layer and other layers in the DR communication architecture. DR information layer represents all the demand response information between the utilities and facilities. DR functional layer describes the information layer's function and DR business layer represents the demand response program under which the facility might participate.

3.2 Mapping DR Pyramid with Automation Pyramid

Since the thesis specifically describes the DR communication between industrial facility and utility. Therefore, we can able to map the DR communication architecture with the traditional industrial automation pyramid as shown in the below Figure 26.

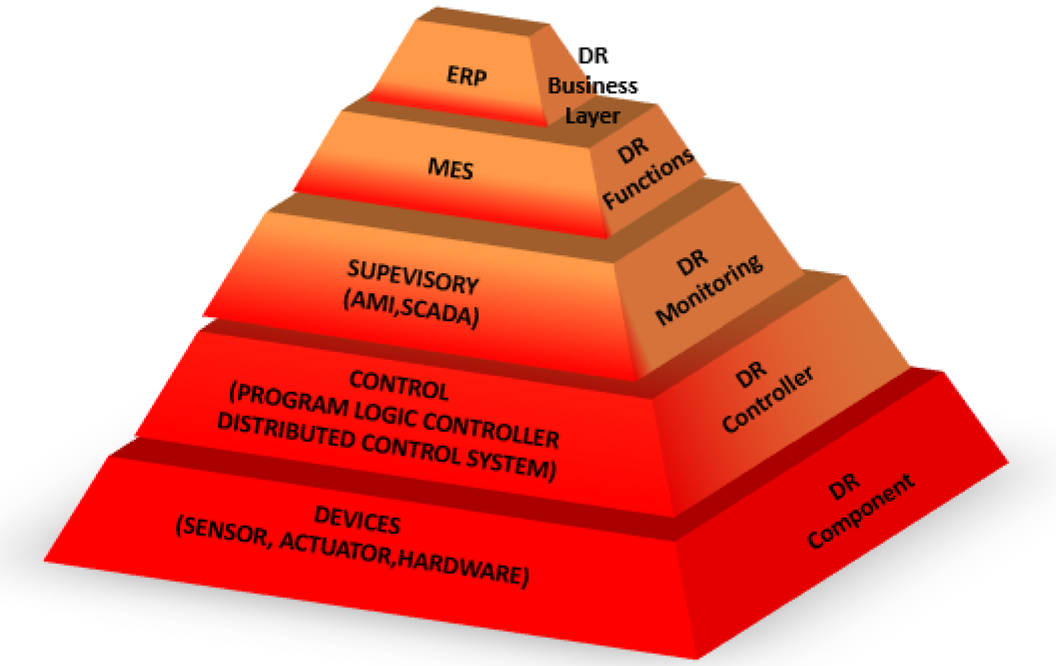


Figure 26. *Mapping between DR communication and automation pyramid*

DR component was explained previously, and they are the resources in the automation pyramid device level. DR controllers are the logical controllers that can able to control the resources and mapped to control level. The supervisory level is mapped with DR Monitoring which monitors the energy consumption in the industrial facility. Like MES functions, Demand Response have their own set of functions that are generic and open. These functions are used to collect resource electricity consumption data and to scheduling facility resource depending on their DR program. DR business layer involves managing the transactive service in demand response interaction.

For an effective demand response program, the industrial facilities should be integrated to the existing facility infrastructure with the demand response infrastructure.

3.3 Categorizing DR participants

Categorizing the DR participants involves differentiating them based on OASIS Energy Interoperation naming convention. During categorizing phase, the application developer must decide roles of the demand response participants and this is out of scope of specifications. Therefore, the participants involved in DR program are categorized into VEN's and VTN's respectively. The VEN's and VTN's should be identified, grouped and categorized proportionally to the number of equipment's resides in the DR facility. The grouping of facility resource based on their energy consumption pattern is out of scope of this thesis. Therefore, the facility resources as grouped under Factory Energy Management System (FEMS) and Building Energy Management System (BEMS).

Under FEMS, any physical devices that consumes electricity in the production line which is responsible for any process to manufacture a product are grouped. Under BEMS, any physical device consuming electricity that supplements the production line in the process of manufacturing products are grouped.

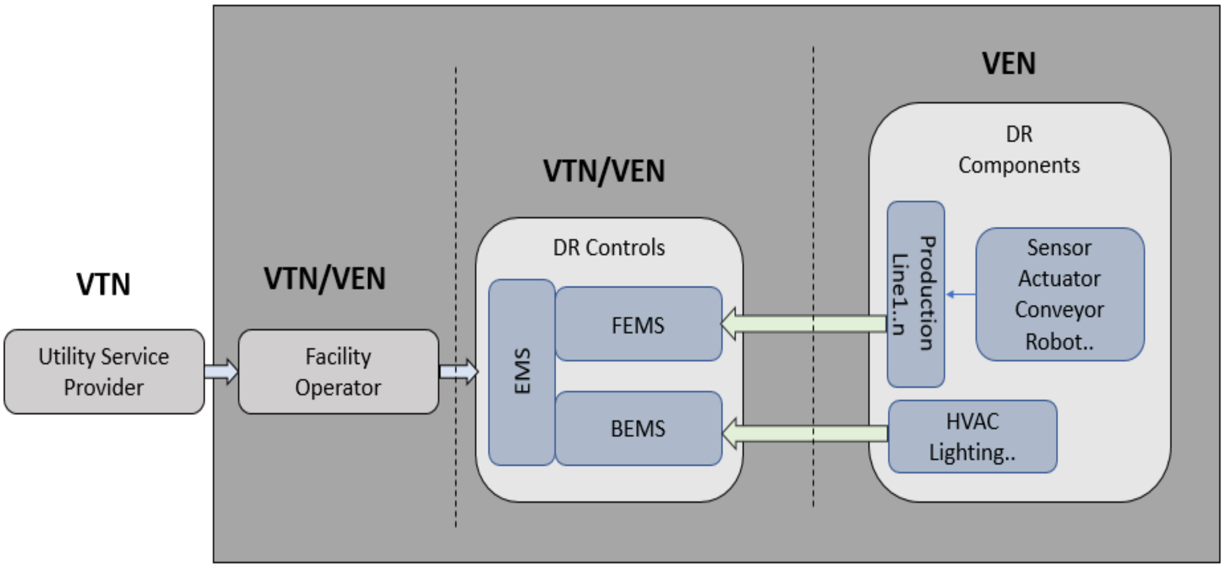


Figure 27. *DR Categorization*

The above Figure 27 describes the DR categorization of demand response program participants based on OASIS EIOp convention. The DR components are the Virtual End Nodes in the demand response program. They curtail their electricity usage in the demand response program. DR controls and Facility Operator (FO) can either be Virtual Top Node or Virtual End Node depending on their action in DR program. Since we excluded the energy market context, Utility Service provider is considered as the Virtual Top Node in this thesis.

3.4 Modeling DR Functions

The VEN and VTN interaction in the demand response program is represented in use case diagram as show in the below Figure 28.

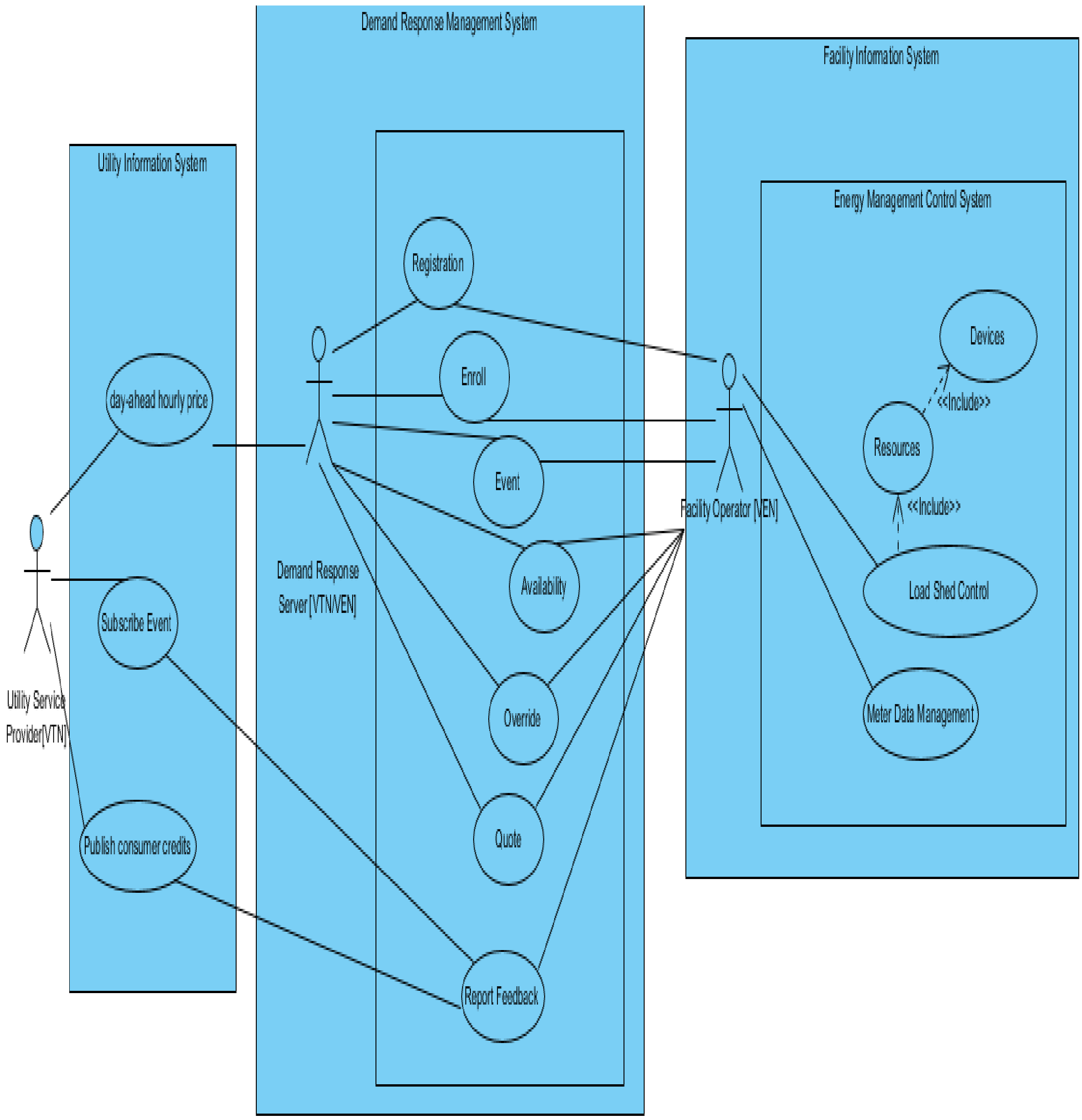


Figure 28. *Demand Response System Use Case*

The description of above use case diagram is given in the following Tables 10, 11 and 12. Each table will have system specific descriptions.

Table 10 describes the system of the demand response program.

Term	Acronym	Description
Utility Information System	**UIS**	The scope of the system is boundless but limited for DR program actors. The system represents functionality that it will provide day ahead hourly price, subscribe and publish a few DR events.
Demand Response Management System	**DRMS**	The scope of the system is boundless but aid only for Demand Response Server (DRS). The system may include functionality for forecasting and prediction of DR events which supports for customer's decision making but not limited to only those. DRS is used to facilitate the facilities replies to various Demand Response services. The database consists of all the historical event information for the utility and the facility site.
Facility Information System	**FIS**	The scope of the system is boundless but limited to the functionality for only Demand Response Events in the facility site.
Energy Management Control System	**EMCS**	The system represents the functionality for monitoring and controlling the energy consumption by the resources not specifically when DR event initiated but also serves for other purpose which is out of scope.

Table 10. *System and their description*

Table 11 describes the actor's role in the demand response program.

Term	Acronym	Description
Utility Service Provider	**USP**	Responsible to deliver demand response. They also coordinate the Energy Market Information System with outside environment.
Demand Response Server	**DRS**	A role which is bridge between all the possible interactions between VTN and VEN's in the demand response program.
Facility Operator	**FO**	A role which directly acts on the resources for load shedding or load shifting when DR signals are initiated. This includes all the energy management related activities in the facility site.

Table 11. *Actor's and their description*

Table 12 below describes the use case in the demand response program.

Use Case	Description
Day-ahead hourly price	Quotes day-ahead price of electricity once the price date is published.
Subscribe Event	Reports generated from the facility are saved and published to utility operator.

Publish Consumer Credits	This is the feedback to the facility
Enroll	The facility site to participate in the DR program.
Event	Events are used for communication between VTN and VEN's and consist of various attributes.
Availability	To view the resources available for the DR program resides in the facility.
Override	A temporarily change of resource availability
Quote	Price distribution to all the VEN's present in the Utility Service Provider list.
Report Feedback	Represents information about the energy consumption form the consumer.
Demand Response Signals	Signal used to invoke the DR event in the facility site.
Load shed Control	Controllers present in the facility which initiates DR event at a pre-defined timestamp.
Resources	Resources may include all the elements which consume energy.
Meter Data Management	Used to monitor overall energy consumption profile in facility.

Table 12. *Use case and their description*

3.5 DR Use Case Information Modeling

A short description of DR Service is explained in Chapter 2.2.1. Information modeling is essential in demand response communication. This modeled information is consumed by all actors involved in each phase of communication. Therefore, the structure of each use case is modeled using class diagram. Each use case has their bounded class, attributes, methods and operation payloads. The modeled class diagram is generic and could be extended depending on the user's requirements. These class diagram and message schemas are taken from OASIS Energy Interoperation version 1.0 standard [39] and modified to this thesis context.

1. *eiRegisteration Service*

eiRegisteration Service will have EiParty Type and EiRegisteration Type classes. The EiParty Type will have information about partyID, partyName, partyRole and partyLocation and EiRegistration Type will have information about registreePartyID, registararPartyID and registrationInfo. The Figure 29 depicts eiRegisteration service class diagram.

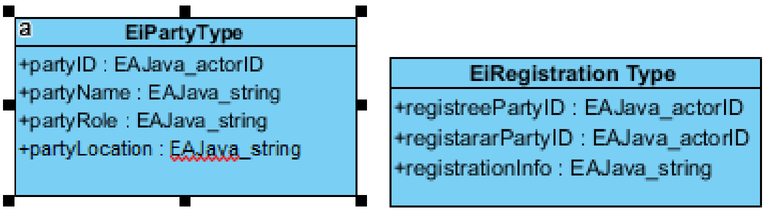


Figure 29. *eiRegisteration service class diagram*

The corresponding XSD (XML Schema Definition) for EiRegisterationType and EiParty, eiRegisteration service operation payload and eiRegisteration service operation payload XDS are given in appendix B – Eiregisteration and eiparty schema's

2. *eiQuotePrice Service*

eiQuotePrice Service will have EiQuote Type and EmixBaseType classes. The EiQuote Type will information about day-ahead electricity price quoteID and inherits properties from EmixBase Type. EmixBase Type will have information related to electricity market context. The Figure 30 depicts eiQuotePrice service class diagram.

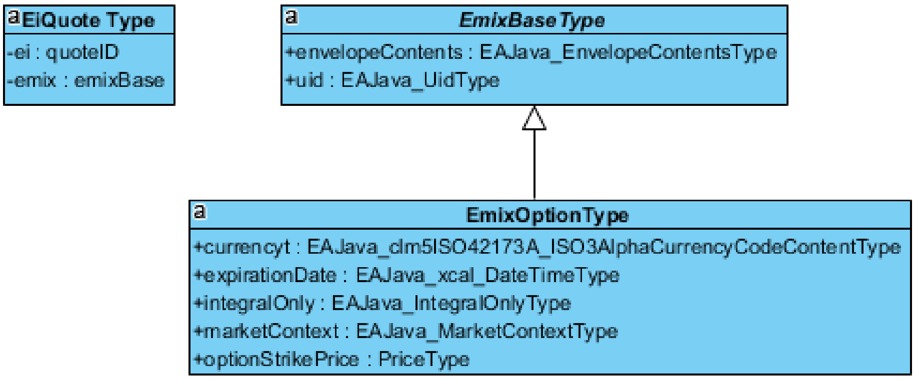


Figure 30. *eiQuotePrice service class diagram*

The corresponding XSD for EiQuote, eiQuote service operation payload and eiQuote service operation payload XDS are given in appendix C – eiquote and emixbase type schema's.

3. *eiAvailability Service*

eiAvailability Service will have EiAvail Type and EiAvailBehavior enumeration classes. The EiAvail Type notify the VTN about the resource availability in the facility. EiAvail uses EiAvailBehaviour Type for the VEN to respond to VTN.

Accept – To accept the load curtail DR Event issued by VTN

Reject – To reject the load curtail DR Event issued by VTN

Restrict – To modify load DR Event issued by VTN.

The Figure 31 depicts eiAvailability service class diagram.

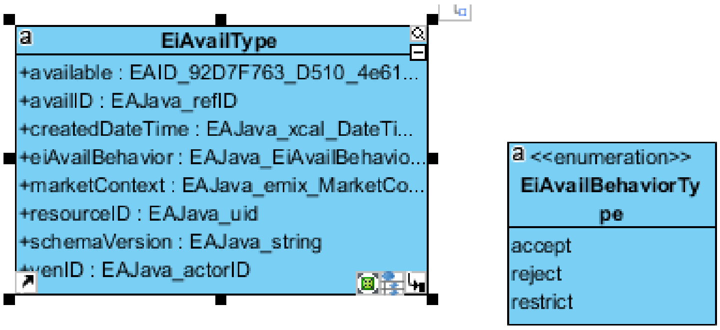


Figure 31. *eiAvailability class diagram*

The corresponding XSD for EiAvailablity, eiAvailablity service operation payload and eiAvailablity service operation payload XDS are given in Appendix D – eiavailablity schema.

4. eiOverride Service

eiOverride Service is a temporary event in which the VEN's may either OptIn or OptOut in the curtailment process without affecting the eiAvailability Service. The Figure 32 depicts eiOverride service class diagram

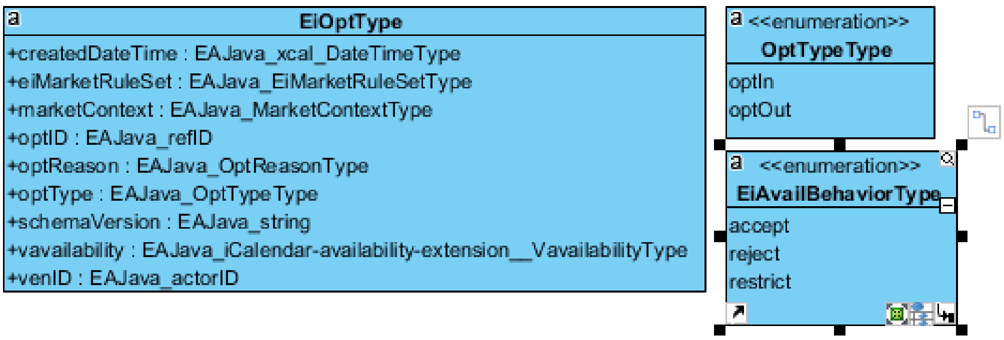


Figure 32. *eiOverride class diagram*

The corresponding XSD for EiOverride, eiOverride service operation payload and eiOverride service operation payload XDS are given in appendix E – eioverride schema

The class diagram, the operation payload and message schema for eiEnrollement Service, eiEvent Service and eiReport Service are given in APPENDIX F – eienrollement schema, appendix G – eievent schema, appendix H – eireport schema.

3.6 DR System Communication Model

After modeling the use case information, we need to determine the procedural communication between the demand response systems. The activity diagram shown in Figure 33 below describes the behavior between Utility Information System (UIS), Demand Response Management System (DRMS) and Facility Information System (FIS).

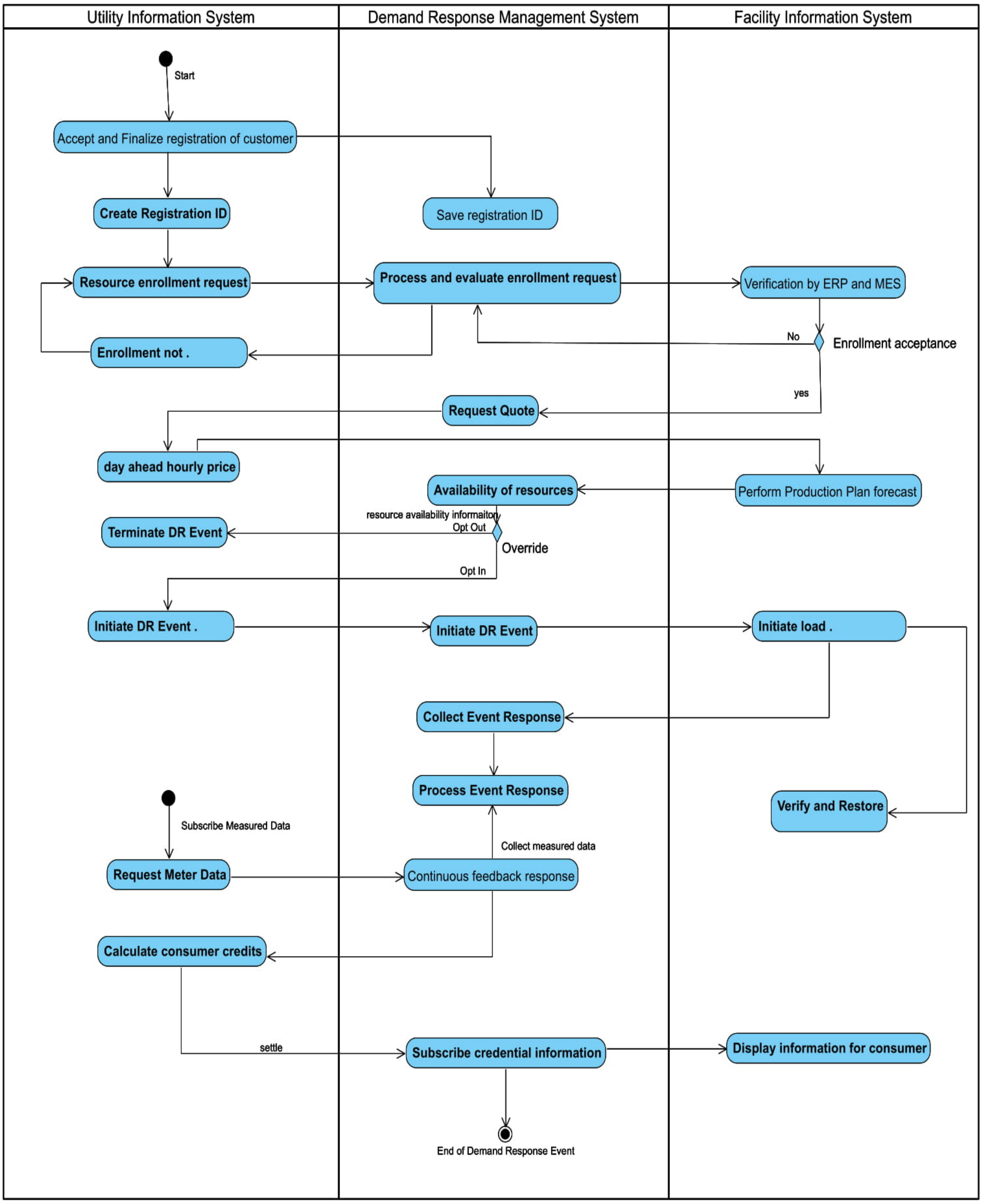


Figure 33. Demand Response system communication

3.7 DR Communication Sequence

After identifying the essential information to be communicated between the VEN and VTN and modeling the system communication model, we need to model the order of communication flow between the participants. Figure 34 depicts the sequence diagram defining the order of EiService communication between the demand response participants. The diagram does not show the interaction of operation payloads since it will be described while implementing those service in Chapter 4.

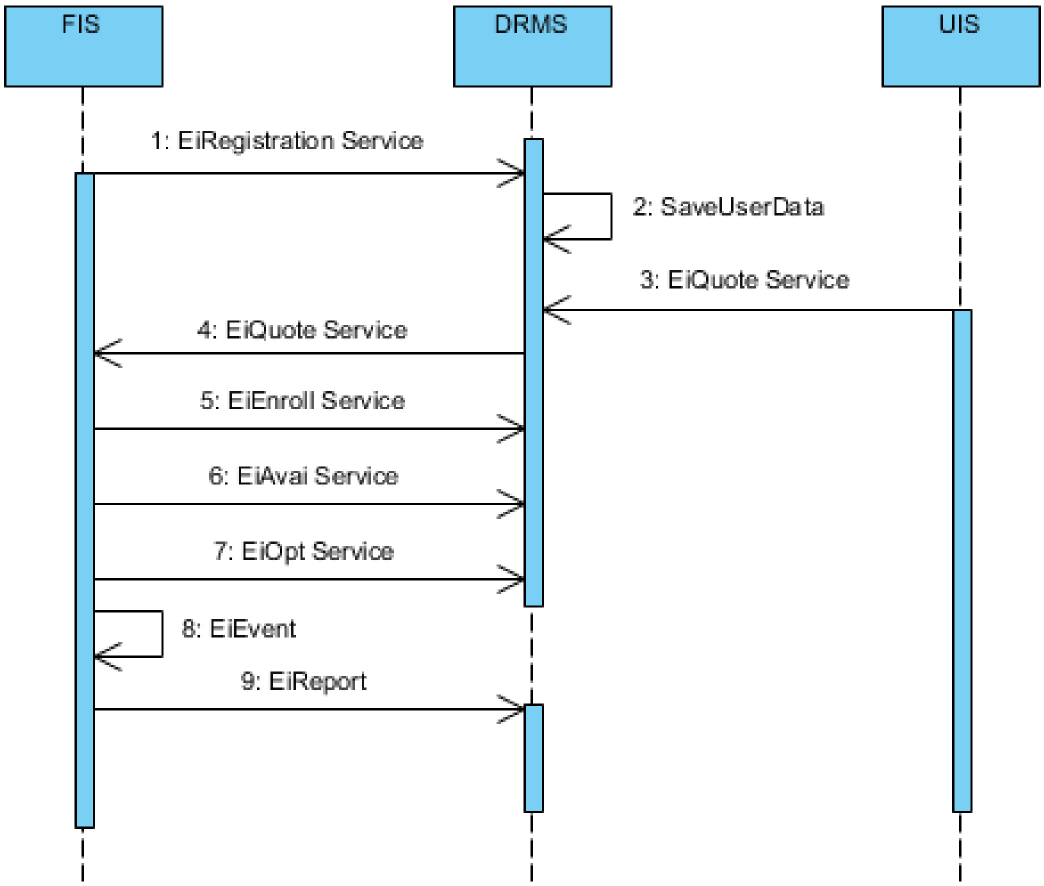


Figure 34. EiService communication sequence

3.8 DR Communication Technology

Once we know the communication sequence, we need to choose VEN and VTN's communication technology. The following Figure 35 shows a method to decide between PUSH and PULL depending on the participants role in DR program.

COMMUNICAITON MODEL	SUITABLE PARTIES
PUSH	Suitable for the utilities (VTN's) to reach many participants in the demand response program
PULL	Suitable for the facilities (VEN's) to get only the required EiService from the utilities
PUSH/PULL	Suitable for the participants which acts both as VTN/VEN

Figure 35. *Choosing DR communication technology*

3.9 Frameworks and Tools

3.9.1 Liferay IDE

Liferay Integrated Development Environment (IDE) is an extension of Eclipse IDE used to develop portal application. Liferay is built on java platform. Liferay IDE provides persistence of data using service builders. With the help of service builders in liferay we can generate web service for our application. The IDE helps us to easily create, customize and manage our portlet in our web application [70]. In this thesis we are using Liferay 7.0 to develop our application. Figure 36 shows liferay workspace in Eclipse IDE.

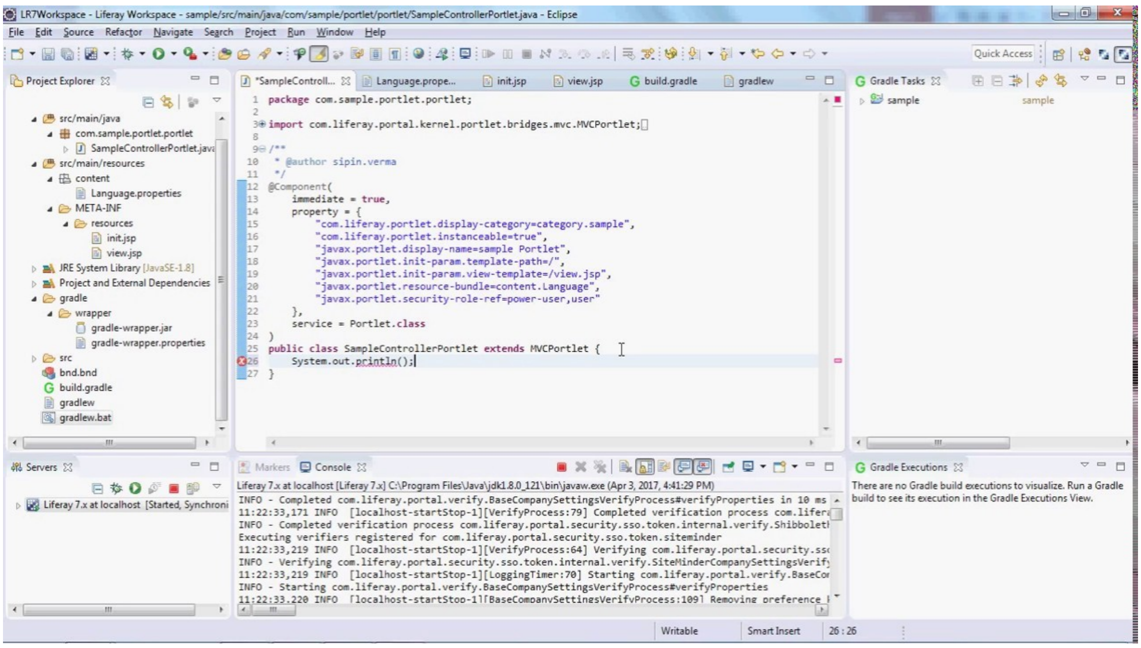


Figure 36. *Liferay workspace in Eclipse IDE*

3.9.2 Apache Maven in Liferay

The build process of application is done using Apache Maven. Maven helps in building the application fast after the development. The below Figure 37 shows the project structure created using maven tool.

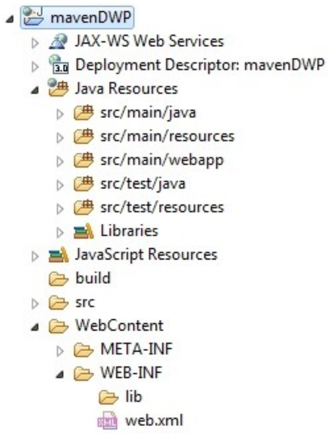


Figure 37. *Maven project structure*

The projects in Maven are configured and managed using Project Object Model (POM). They are stored in xml file type. A simple project structure of maven pom.xml file is shown below:

```
<project>
  <!-- model version is always 4.0.0 for Maven 2.x POMs -->
  <modelVersion>4.0.0</modelVersion>

  <!—project description -->

  <groupId>group ID</groupId>
  <artifactId>artifact ID</artifactId>
  <version>P</version>

  <!-- library dependencies -->
  <plugins>
        <dependencies>
                <dependency>

                 </dependency>
        </dependencies>
  </plugins>
</project>
```

The POM allows us to configure various JAR’s (Java Archive) required for our project dynamically and provides transparent information about the JAR’s. The maven dependencies help the user to have version control over the JAR’s used in our project. A simple maven dependency for liferay faces is shown below [71]:

```
<!--https://mvnrepository.com/artifact/com.liferay.faces/liferay-facesbridge-
impl -->
<dependency>
    <groupId>com.liferay.faces</groupId>
    <artifactId>liferay-faces-bridge-impl</artifactId>
    <version>4.2.5-ga6</version>
</dependency>
```

3.9.3 Service Builder

Liferay uses service builder to define the business logic for an application. Object-Relation Mapping (ORM) technology is used by the service builder to distinguish object model and database. The application developer provides a xml file as an input to the service builder. Based on the xml file the object model, persistence and service layer are created automatically. For the application using database, CRUD (Create, Read, Update and Delete) operation code is generated by the service builder. A simple service builder is shown in Figure 38 below:

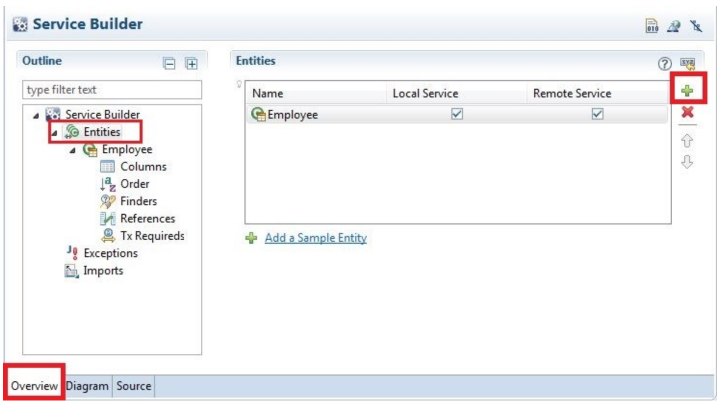


Figure 38. *Service builder*

3.9.4 Apache Tomcat

Apache Tomcat bundle is used as the application server for Liferay portal. Apache Tomcat is a Java Servlet implementation and have web socket technology that supports HTTP connection. For our thesis we will be using tomcat server 8.0.32. The Figure 39 shows the apache tomcat server folders.

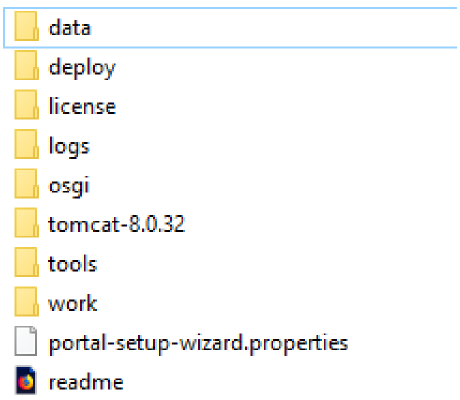


Figure 39. *Apache tomcat server folder*

The data folder consists of default configuration files when liferay is started. The liferay portal configuration files are in the tomcat-8.0.32 folder. The portal dependent JAR's are in that folder.

3.9.5 Tomcat Database Configuration

The liferay portal needs to handle the data source provided by the user. Therefore, it is essential to configure one of the pre-defined databases provided by the liferay. For the portal to access the database, we need to configure the *portal-setup-wizard.properties* from the figure above. The configuration file as below:

```
jdbc.default.driverClassName=database driver
jdbc.default.password=password to access the database
jdbc.default.url=database address
jdbc.default.username=username to access the database
liferay.home=Tomcat server location
setup.wizard.enabled=used to override the default access to the por-
tal(true/false)
```

A detailed description about implementation of these methodological approaches with the tools is presented in Chapter 4.

4. IMPLEMENTATION

Following the methodological approaches discussed in previous Chapter 3, this chapter discusses the implementation of methodology. The chapter begins by describing the facility resources used for demonstrating the demand response communication infrastructure. After that four states for implementing the demand response service is described. Then each stage is developed and their information exchange sequence from their information model is explained. Finally, the states are implemented in different portlets is elaborated.

4.1 Facility Resource Implementation

For demonstrating, we will consider a conceptual facility site. The facility site is an actual production line consisting of 12 cells as shown in Figure 40. The facility is located at Factory Automation Systems and Technologies Laboratory (FAST-Lab) in Tampere University of Technology. More description related to the facility can be found from this work [72]. Generically, each cell consists of robots, controllers and conveyors. The cells are controlled with S1000 controller and the energy consumption is monitored using E10 analyzer. The energy consumption monitoring infrastructure have been studied and deployed in the facility [60]. This thesis partially uses those monitoring techniques for the conceptual facility.

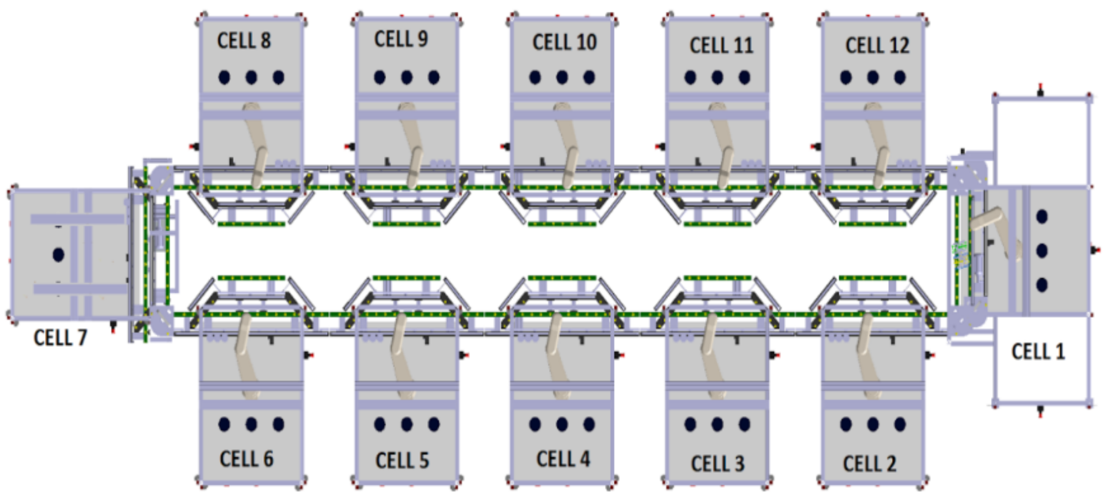


Figure 40. *Fastory production line*

However, we will be considering cell 1 to 3 for the demonstration of demand response states in the following sections.

4.2 DR Service States

The implementation of the demand response service is carried out in four states in sequential manner as shown in the below Figure 41.

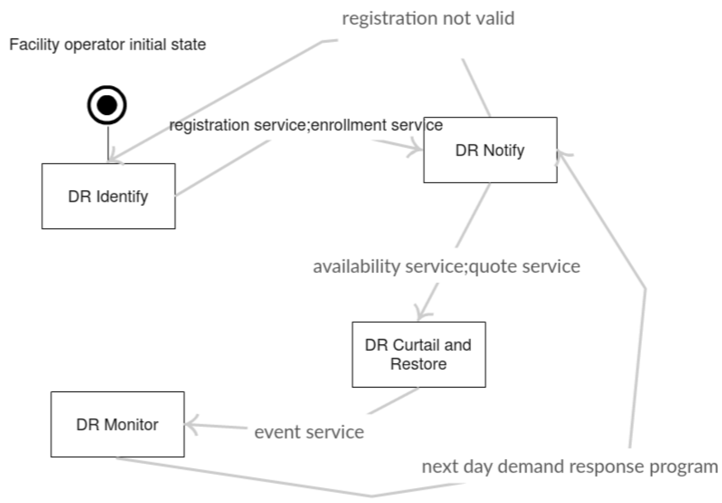


Figure 41. *Demand Response service states*

The first state is DR Identify, the second state is DR Notify, the third and final states are DR Curtail and Restore, and DR Monitor. The implementation of these states are explained in the following sub-chapters.

4.3 Implementation of DR Services

Each demand response service states are developed as a portlet application in this thesis work. The communication between the conceptual demand response actors are represented using the sequence diagram. The presentation for each services are also explained for the respective services,

4.3.1 DR Identify

The Demand Response program is initiated in the DR Identify state. This state provides two services namely, DR Registration Service and DR Enrollment Service.

DR Registration Service:

Figure 42 represents the registration information flow between the Utility, Demand Response Management System, and Facility Information System. The VEN – FIS uses DR Registration service to register with the VTN- DRMS. The registration is initiated by the VEN requesting for the available types of DR program. VTN -Utility replies with registration information.

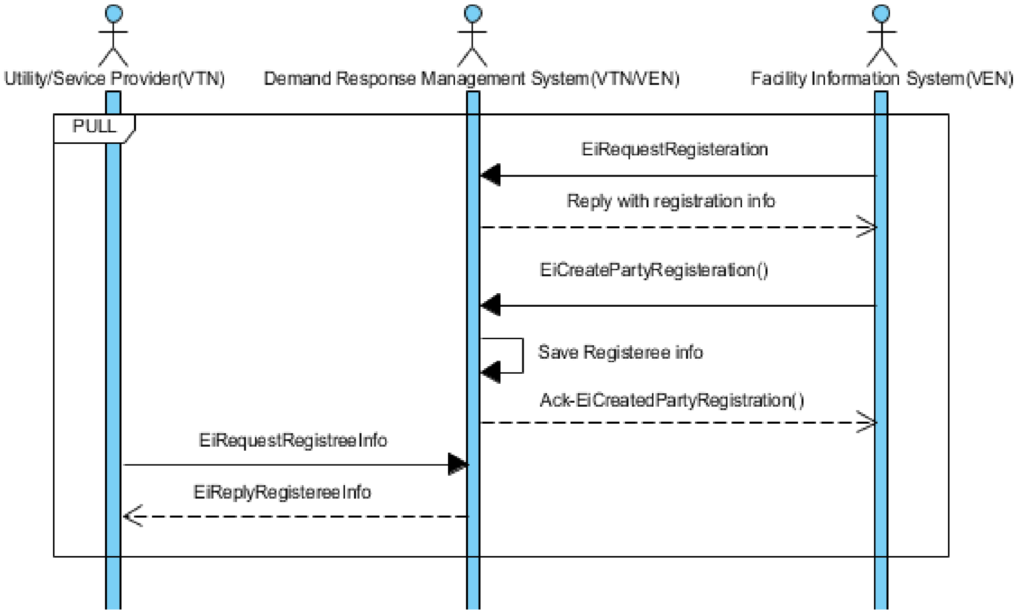


Figure 42. *Interaction Diagram: DR Registration Service*

VEN is required to fill the information and sends the registration form to VTN. The VTN saves the data and groups them accordingly with other available VEN's. VTN creates an ID for the registered VEN and send an ID as an acknowledgement. The ID is used for updating and cancelling the VEN registered information.

If an error occurred during the registration, the VTN sends an error code instead. Once the VEN's information is available in the DRMS, the Utility can request and verify the Facilities information.

DR enrollment Service:

Figure 43 represents the enrollment information flow between DRMS, FIS and Facility Resource. The enrollment is different from the registration. The resources present in the facilities are identified and enrolled using the enrollment service. During the enrollment, the facility operator enrollees the resources to the Factory Information System. First, the operator checks the resource status. Then, he/she enroll the resource to the demand response management system.

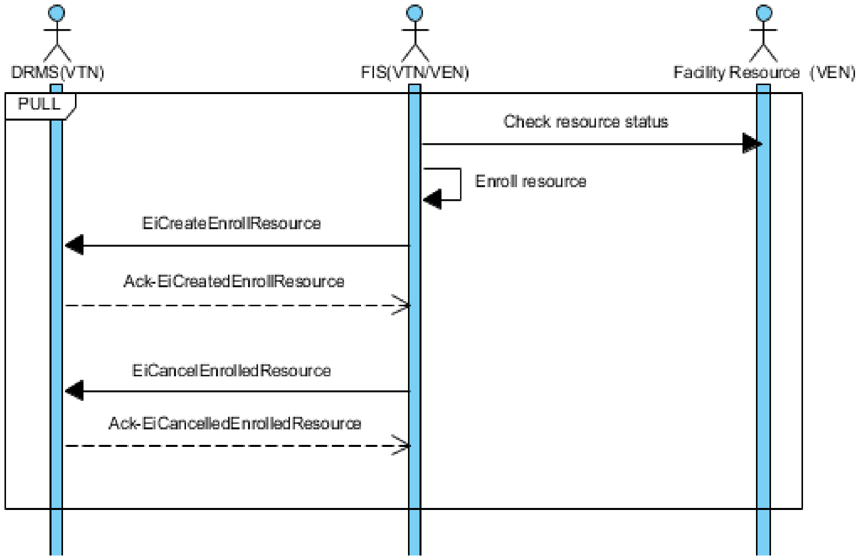


Figure 43. *Interaction Diagram: DR Enrollment Service*

The operator has all the rights to cancel any enrolled resources. Each enrolled resources have unique ID's. While enrolling the resources to the DRMS, the energy consumption data will be added with it.

4.3.2 DR Notify

After registering the DR participants and enrolling the resources to the DR program, the next step is to notify the day-ahead electricity price using DR quote service. The facility operator need to notify the resource availability to take part in the DR program on the day of demand response. Therefore this stage also includes DR Availability and DR Override service.

DR Quote Service:

Figure 44 represents the electricity price information between utility, DRMS and FIS. After the day-ahead market price is available with the utility, the DRMS requests for the price. In reply, the utility creates the electricity price data and send them to DRMS. Either the DRMS can distribute the quoted price to all the registered VEN's or they can publish only to the subscribed VEN's.

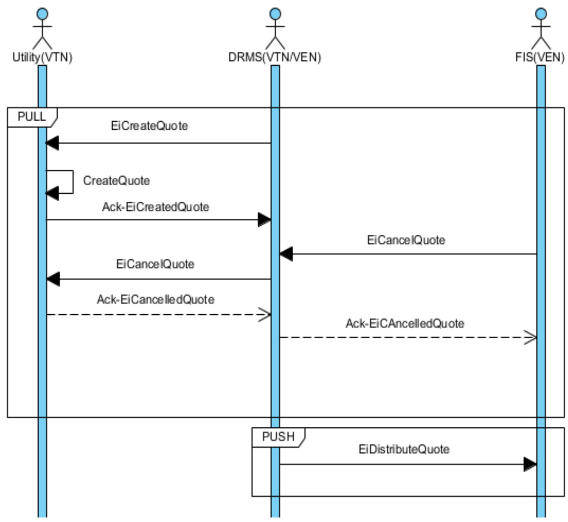


Figure 44. *Interaction Diagram: DR Quote Service*

After receiving the day-ahead market price from DRMS, the VEN's can either accept or cancel the received quote. If a VEN cancels the quote, they cannot participate in the next day's demand response program and vice versa.

DR Availability Service:

Figure 45 represents the facility resource availability information flow between DRMS and FIS. The availability service uses PULL technology. The FIS creates the facility resource availability with the DRMS. The DRMS replies to FIS with an acknowledgement. This helps the DRMS to know the available facility resources that are available for demand response program. The FIS can also cancel any resources from the demand response program with cancel availability event.

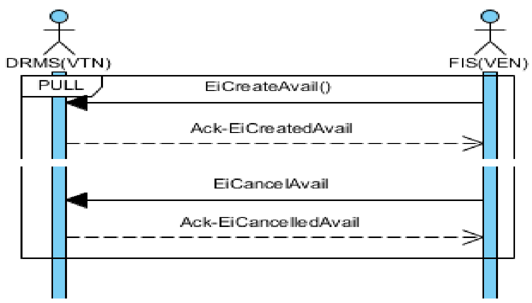


Figure 45. *Interaction Diagram: DR Availability Service*

DR Override Service:

Figure 46 represents the resource's override information flow between DRMS and FIS. Once the FIS chooses Opt-In, the resource must participate in demand response program.

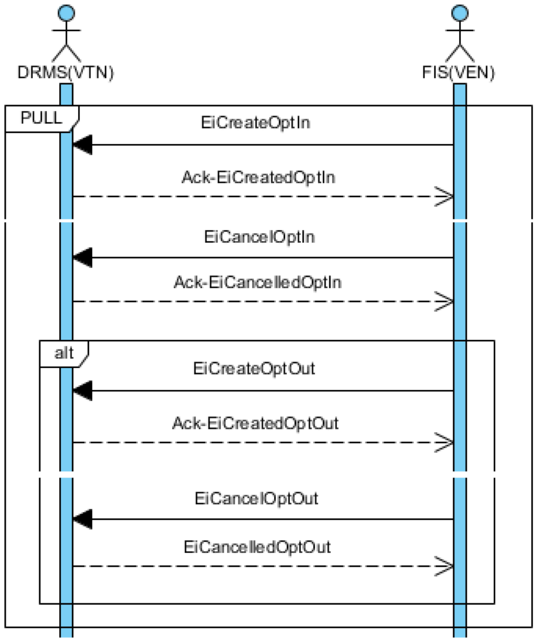


Figure 46. *Interaction Diagram: DR Override Service*

Even tough, the FIS shows availability for a resource, an Opt-Out will make the resource not accepting the demand response event. The FIS can choose the reason for opting-in or opting-out. The reason could be either 'emergency' or 'mustRun'.

4.3.3 DR Curtail and Restore

Figure 47 represents the DR Event Service information flow between the DRMS, FIS and Facility resources. The DRMS initiates the actual electricity curtailment event by sending a create event signal. The FIS can either OptIn or OptOut for the particular event. Once the FIS opted-in for the event, the DRMS distributes the curtailment event. The curtailment event have three periods namely RampUp, Duration and Recovery periods. First, during RampUp period the facility resources starts to decrease its energy consumption. The information about the RampUp period is provided to FIS.

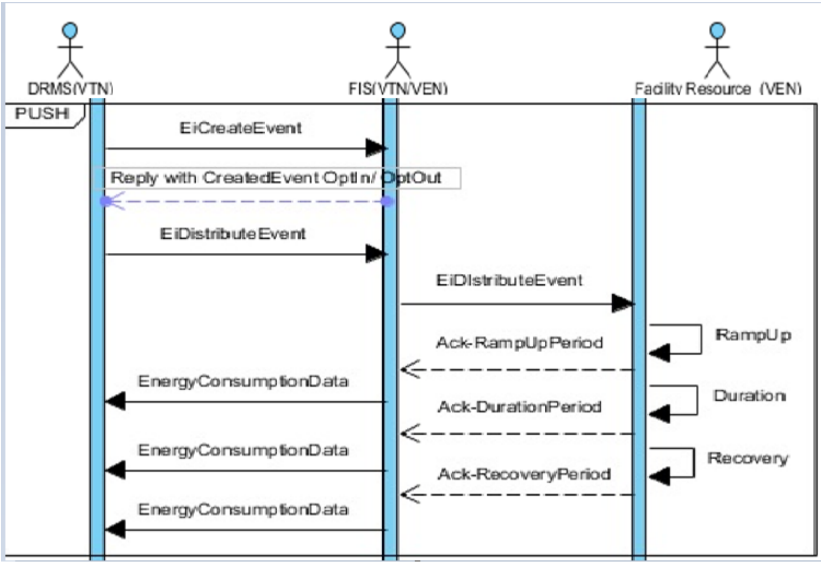


Figure 47. *Interaction Diagram: DR Event Service*

Next, during Duration period the facility resource reduces its energy consumption below their baseline consumption. The information about the Duration period is provided to FIS. Finally, during Recovery period the facility resources retains to their actual energy consumption. The information about the Recovery period is provided to FIS. The energy consumption data for all the periods are sent to DRMS for further demand response services.

4.3.4 DR Monitor

Figure 48 represents the DR Report service information flow between Utility, DRMS and FIS. The FIS requests DRMS to create the demand response event report. The DRMS creates the report.

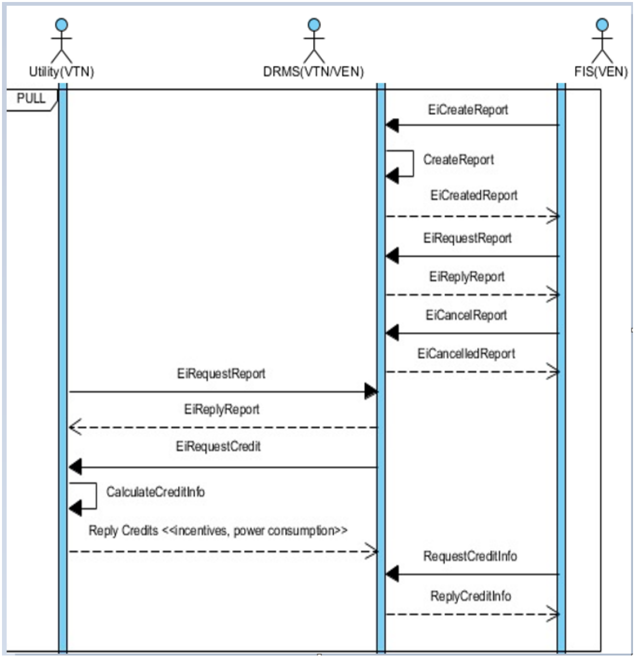


Figure 48. *Interaction Diagram: DR Report Service*

The DRMS replies to FIS and Utility, whenever they request for the demand response report. The Utility prepares credits for the facilities depending on their contribution to demand response participation. These credits are then sent to DRMS which could be requested and accessed by FIS.

5. RESULTS

This chapter shows the user interface of the various DR services discussed in the previous chapter. The result are conceptual within this thesis context and could be used as proof of concept for other manufacturing facilities. Each state of implementation will have their own portlet's where the facility operator can visualize the demand response program. This chapter answers the problem statements given in Chapter 1 and explains how the communication proposed infrastructure meet the requirements.

5.1 DR Registration

Figure 49 shows the conceptual model of DR registration form. The form have three input text fields, where the facility user enters their name. Then, they can choose their facility type from the available check box. The forms allows the user to enter the facility location with their area pin code. Once the details are filled, the user can submit the registration form.

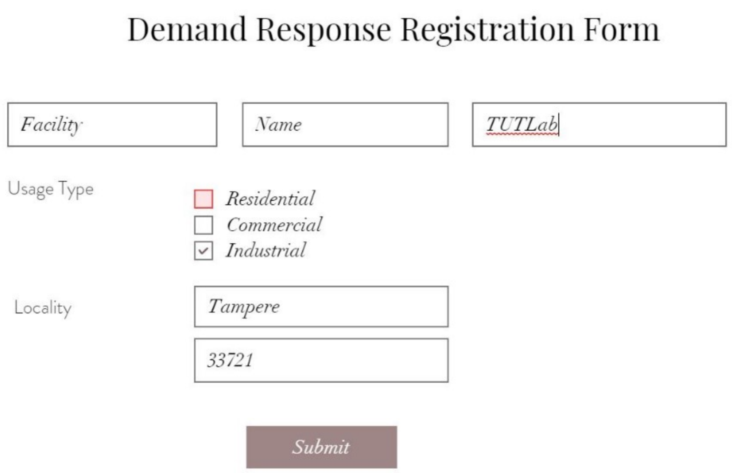


Figure 49. *DR Registration Form*

Figure 50 shows the Home tab of the Facility portal. This displays the information of the facility site. The facility operator can able to cancel their registration from here.

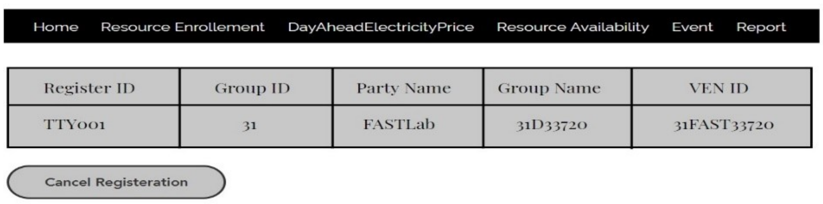


Figure 50. *Home portlet*

5.2 Resource Enrollment

Figure 51 below shows two portlets for resource enrollment. The facility operator can choose the resources from "Enroll Resource" portal and click "Enroll" to enroll the resource in DR program. The "Enrolled Resource" portal shows the enrolled resources and the operator can cancel the enrolled resources from here.

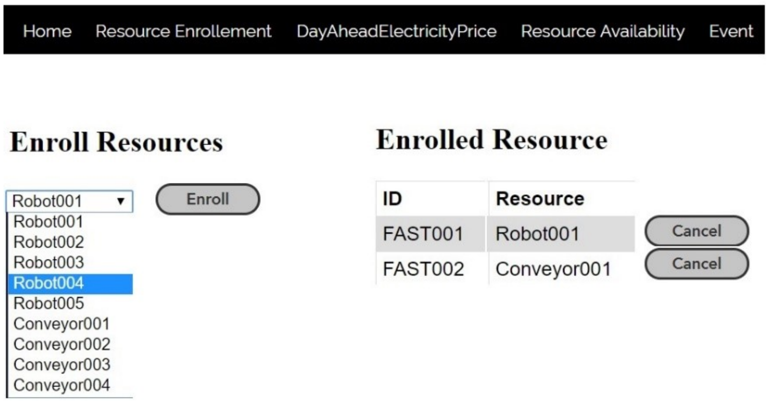


Figure 51. *Enrollment portlets*

5.3 Day Ahead Electricity Price

In this thesis, it was assumed that the facility operator accepted the quoted price. Figure 52 and 53 below shows the day-ahead price in chart view and table view depending on the operators convenient.

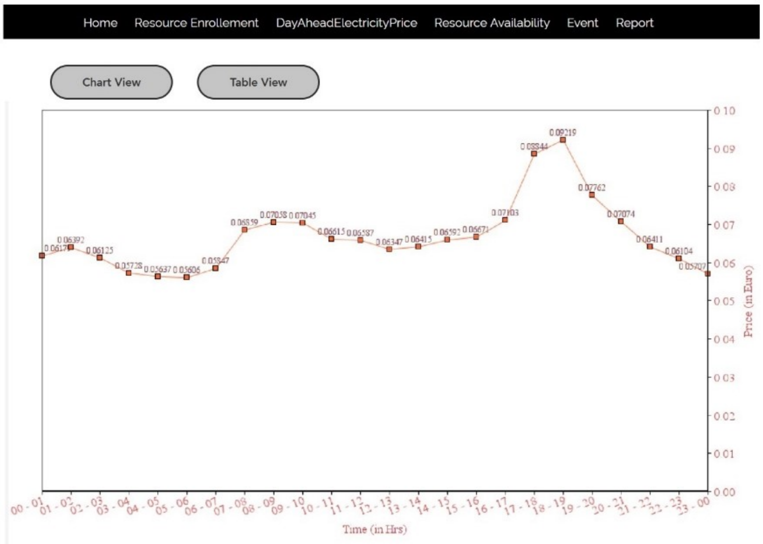
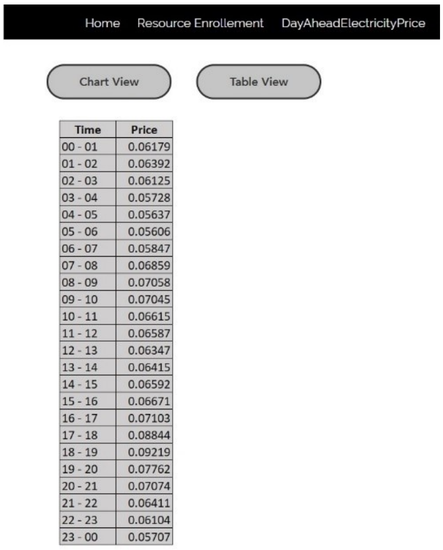


Figure 52. *DR price portlet – Chart view*



Time	Price
00 - 01	0.06179
01 - 02	0.06392
02 - 03	0.06125
03 - 04	0.05728
04 - 05	0.05637
05 - 06	0.05606
06 - 07	0.05847
07 - 08	0.06859
08 - 09	0.07058
09 - 10	0.07045
10 - 11	0.06615
11 - 12	0.06587
12 - 13	0.06347
13 - 14	0.06415
14 - 15	0.06592
15 - 16	0.06671
16 - 17	0.07103
17 - 18	0.08844
18 - 19	0.09219
19 - 20	0.07762
20 - 21	0.07074
21 - 22	0.06411
22 - 23	0.06104
23 - 00	0.05707

Figure 53. *DR price portlet – Table view*

These price data are dynamic and updated every day. The updated price today is tomorrow's electricity price for the facility consumption.

5.4 Resource Availability

Resource availability portlets are shown in Figure 55. Firstly, the operator needs to accept or reject the event. And depending on the event the facility operator can choose the resources from the Resource Name and their available Start and End time to create the availability. The operator can add any number of available resources and can also cancel if needed.

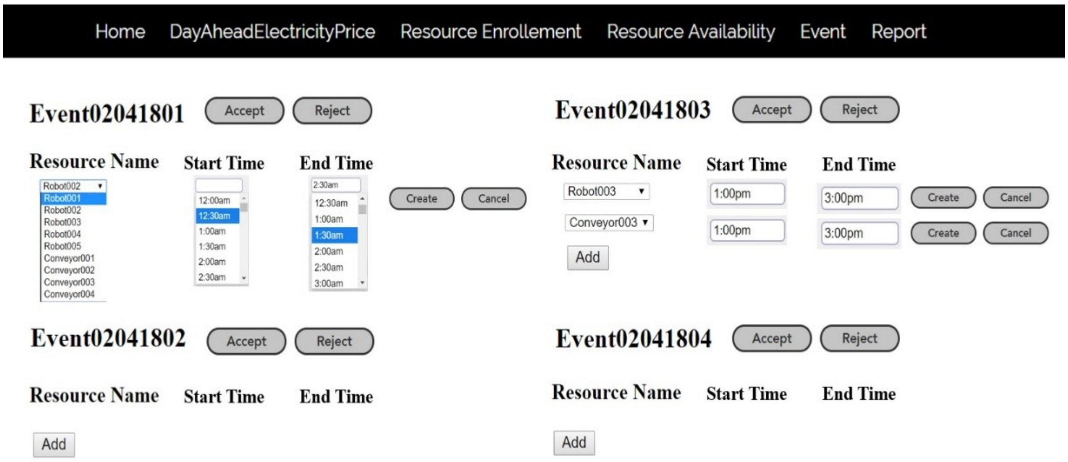


Figure 54. *Resource availability portlets*

5.5 Event

Figure 55 shows the event portlets. The Event Override portlets allows the user to select the different event. Based on the event, the resources are displayed and chosen from there. The facility operator must select OptIn or OptOut for the selected resources and accept the demand response event. If the operator chooses OptOut, then they must select the reason from the dropdown.

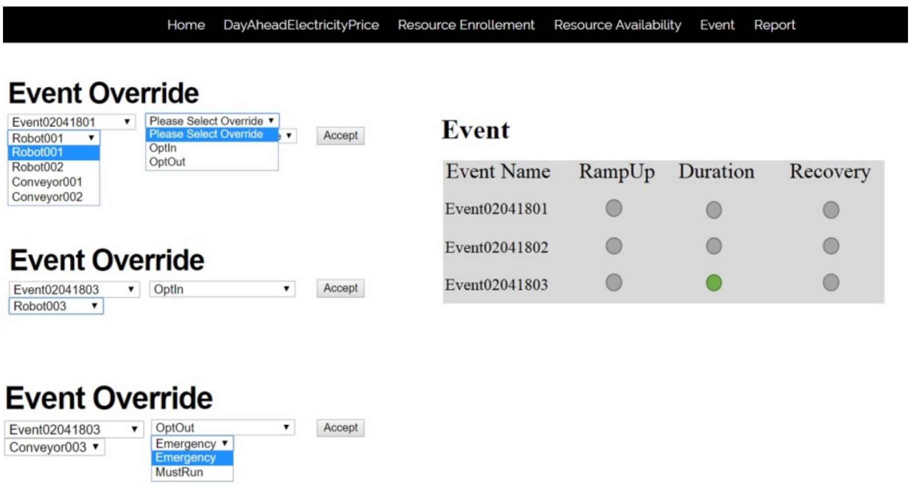


Figure 55. *Event portlets*

The Event portlet shows the ongoing demand response event. Based on the status the operator will have an overview of the resource status.

5.6 Report

Figure 56 shows the visualization of demand response report to the facility operator.

Figure 56. *Report portlet*

The historical energy consumption data of the resource are used us reference for plotting the report. The report shows the overall demand response event for all the resources participated in the demand response program. For future implementation, this report can be modularized for each event separately and a portlet for each event.

5.7 Concepts and learnings

As an end product from this thesis work, an approach and suggestion for problem defined in Section 1.2.2 has been designed and developed. The solution stated below responds to the problem statement.

- ***What are the types of demand response program and which is suitable for the user?***

There are different types of demand response program as mentioned in Chapter 2.1.1. It is entirely the facilities decision to choose the right program that fits them. In the thesis, Real Time Pricing (RTP) program was selected. Because firstly, the facility is a factory floor where all the resources have their scheduled time to do their own task. Since RTP electricity price varies hourly, the resources can easily be scheduled to any of the hours for demand response program. Secondly, the operator receives the hourly price a day ahead. This helps the facility operator to plan their production effectively. Finally, RTP program is flexible compared to other programs meaning the facility operator have full control of demand response program.

- ***What are the demand response standards and how to choose the standard depending on one's need?***

The standards were discussed in Chapter 2.2. And choosing the standard depends on the facilities requirement. Nowadays, OpenADR is most widely used demand response standard. However, in this thesis we used OASIS Energy Interoperation which is more open. This is because, OASIS is the core of demand response standard and we can filter and develop our specification from it. In that specification we can synergize different demand response standard based on our needs.

- ***What is the essential information required from the standard and how this information could be aligned with the facility infrastructure?***

The standard provide detailed information to develop the demand response program. Since there were some limitations as stated before, the essential information was conceptualized with different demand response services. The functions of these services were aligned with the automation pyramid. This was generic and this could be used as a proof of concept while modeling the demand response program for other facility infrastructures.

- ***How this information is sequenced between the demand response participants and which communication model is used to transmit the information?***

The flow of demand response service must be sequenced so that the facility operator understands the demand response step-by-step. Therefore, states for the services were explicated in the thesis. Also the communication model depends on the demand response participant's service initialization. If a facility initializes the service PULL method was used and if the utility initializes PUSH method was used. However, this may not suit for all the services. Sometimes the methods may interchanged depending on the convenient of the participants.

- ***How the facility visualize their demand response program?***

The important hindrance for the facility is how to view or visualize their demand response program. To overcome the thesis suggests the use of portals and portlets for visualizing the demand response program. Even though the mentioned portlets does not give all the information regarding the demand response program, essential information are visualized.

6. CONCLUSION AND FUTURE WORK

This chapter gives an overview of the approach and solution achieved on constructing the communication infrastructure for demand response program. This chapter will also discuss the proposed infrastructure and implementation for demand response program. Some possibility of future work is also discussed in this chapter.

6.1 Conclusion

The manufacturing facilities uses electricity as needed but they can reduce their usage below their baseline usage. The acceptance of DR program in manufacturing facilities is few. Even tough, the utilities are encouraging the industrial EC's and willing to give incentives, the EC's are still skeptical on the DR program. This resulted in framing a problem statements described in Section 1.2.2. A methodology was proposed that gives an opportunity for the facility operator to align their manufacturing facility to the demand response program.

The demand response standards were discussed in details and comparisons were made. This will help the facilities to tailor the choice of information of their own. The complete communication infrastructure with different demand response layer was modeled in Chapter 3. After that mapping of these layers with the automation pyramid was synthesized. With the mapping, it will be simple for any new facility to map their existing infrastructure with the proposed one.

The participants for demand response program was identified and classified based on the problem statement. Later, with the help of a use case, the demand response functions were modeled. This modeling helps the facility operator to specifically understand the role of each actors in the demand response program. After modeling the use case, each demand response functions were modeled within the thesis context. Even though it is not simple to understand for the end user, the demand response program developer in the facility can able to further extend those functions.

The Chapter 3.7 explains the demand response communication sequence. The facility operator can refer to the activity diagram to get an overall picture of the demand response program sequence as a whole. During the implementation phase, each demand response services were grouped to execute the DR events in states. These states explain the communication sequence between the DR participants and their interaction between them.

The result obtained from the implementation was explained in Chapter 5. Portlets were developed for each demand response services. The facility operator can configure the portlet easily in the web page without any knowledge of programing. The infrastructure is more generic and the approach can be utilized in manufacturing facilities.

6.2 Future Work

The infrastructure developed as part of this thesis, is in the early stage. The infrastructure must be improved to be flexible and interoperable in future work. Some of those improvements are discussed in this section.

Firstly, the infrastructure is not robust while mapping with the MES system. Therefore, more MES functions must be coupled with the demand response functions. The ISA – 95 standard must consider including the demand response program function. The program must be an integral part of all the manufacturing industry who is accepting industrial revolution 4.0.

The portlet communication is good for limited number of resources. But with the increased number of resource in the facility, transmitting and receiving of information becomes complex. Therefore, IoT communication protocol like CoAP must be considered instead of HTTP. The user interface for demand response program could be much simpler so that anyone can easily learn. Therefore, testing of the user interface is required deploying to the facility.

As a final note, demand response might seems like a simple task to integrate to an existing infrastructure. But in reality implementing the demand response infrastructure require more investment. Some facilities understand the real value in DR program and starts to invest in it. But some does not want to step out from their favorable conditions. The government need to provide incentives to those willing to participate in DR program and more pilot programs are needed to evaluate the programs.

REFERENCES

[1] "Electricity use by machine drives varies significantly by manufacturing industry - Today in Energy - U.S. Energy Information Administration (EIA)." [Online]. Available: https://www.eia.gov/todayinenergy/detail.php?id=13431. [Accessed: 22-Mar-2018].

[2] C.-W. Park *et al.*, "Energy consumption reduction technology in manufacturing — A selective review of policies, standards, and research," *Int. J. Precis. Eng. Manuf.*, vol. 10, no. 5, pp. 151–173, Dec. 2009.

[3] "FERC: Industries - Reports on Demand Response & Advanced Metering." [Online]. Available: https://www.ferc.gov/industries/electric/indus-act/demand-response/dem-res-adv-metering.asp. [Accessed: 23-Mar-2018].

[4] S. Bougain, D. Gerhard, C. Nigischer, and S. Uĝurlu, "Towards Energy Management in Production Planning Software Based on Energy Consumption as a Planning Resource," *Procedia CIRP*, vol. 26, pp. 139–144, 2015.

[5] J. Torriti, M. G. Hassan, and M. Leach, "Demand response experience in Europe: Policies, programmes and implementation," *Energy*, vol. 35, no. 4, pp. 1575–1583, Apr. 2010.

[6] N. O′Connell, P. Pinson, H. Madsen, and M. O′Malley, "Benefits and challenges of electrical demand response: A critical review," *Renew. Sustain. Energy Rev.*, vol. 39, pp. 686–699, Nov. 2014.

[7] N. G. Paterakis, O. Erdinç, and J. P. S. Catalão, "An overview of Demand Response: Key-elements and international experience," *Renew. Sustain. Energy Rev.*, vol. 69, pp. 871–891, Mar. 2017.

[8] "Market data." [Online]. Available: http://www.nordpoolspot.com/Market-data1/Dayahead/Area-Prices/FI/Hourly/. [Accessed: 04-Apr-2018].

[9] "Demand-Side Management," *Fingrid*. [Online]. Available: https://www.fingrid.fi/en/electricity-market/demand-side-management/. [Accessed: 04-Apr-2018].

[10] V. S. K. Murthy Balijepalli, V. Pradhan, S. A. Khaparde, and R. M. Shereef, "Review of demand response under smart grid paradigm," 2011, pp. 236–243.

[11] "Demand Response | Department of Energy." [Online]. Available: https://www.energy.gov/oe/activities/technology-development/grid-modernization-and-smart-grid/demand-response. [Accessed: 14-Apr-2018].

[12] F. Shariatzadeh, P. Mandal, and A. K. Srivastava, "Demand response for sustainable energy systems: A review, application and implementation strategy," *Renew. Sustain. Energy Rev.*, vol. 45, pp. 343–350, May 2015.

[13] "Benefits of Demand Response in Electricity Markets and Recommendations for Achieving Them," p. 122.

[14] O. Abrishambaf, M. A. F. Ghazvini, L. Gomes, P. Faria, Z. Vale, and J. M. Corchado, "Application of a Home Energy Management System for Incentive-Based Demand Response Program Implementation," in *2016 27th International Workshop on Database and Expert Systems Applications (DEXA)*, 2016, pp. 153–157.

[15] H. A. Aalami, M. P. Moghaddam, and G. R. Yousefi, "Demand response modeling considering Interruptible/Curtailable loads and capacity market programs," *Appl. Energy*, vol. 87, no. 1, pp. 243–250, 2010.

[16] P. Cappers, J. MacDonald, and C. Goldman, "Market and Policy Barriers for Demand Response Providing Ancillary Services in U.S. Markets," p. 61.

[17] Dong-Min Kim and Jin-O Kim, "Design of Emergency Demand Response Program Using Analytic Hierarchy Process," *IEEE Trans. Smart Grid*, vol. 3, no. 2, pp. 635–644, Jun. 2012.

[18] J. Saebi, H. Taheri, J. Mohammadi, and S. S. Nayer, “Demand bidding/buyback modeling and its impact on market clearing price,” 2010, pp. 791–796.
[19] Shen Zhao and Zhou Ming, “Modeling demand response under time-of-use pricing,” 2014, pp. 1948–1955.
[20] P. Cappers, A. Todd, and C. A. Goldamn, “Summary of Utility Studies: Smart Grid Investment Grant Consumer Behavior Study Analysis,” LBNL--6248E, 1171525, May 2013.
[21] M. A. R. Muzmar, M. P. Abdullah, M. Y. Hassan, and F. Hussin, “Time of Use pricing for residential customers case of Malaysia,” 2015, pp. 589–593.
[22] M. Ambrosio, “Assessment of Customer Response to Real Time Pricing,” p. 23.
[23] P. Ponnaganti, J. R. Pillai, and B. Bak-Jensen, “Opportunities and challenges of demand response in active distribution networks: Opportunities and challenges of demand response,” *Wiley Interdiscip. Rev. Energy Environ.*, vol. 7, no. 1, p. e271, Jan. 2018.
[24] M. Miner, “Wholesale Electricity Markets.” .
[25] “Introduction to electricity markets, its balancing mechanism and the role of renewable sources.” .
[26] “The main arena for trading power.” [Online]. Available: http://www.nordpoolspot.com/the-power-market/Day-ahead-market/. [Accessed: 03-May-2018].
[27] “European Demand Response (DR) Market, 2018–2025.” [Online]. Available: https://cds.frost.com/p/56579/#!/ppt/c?id=MDDC-01-00-00-00.
[28] S. P. Burger and M. Luke, “Business models for distributed energy resources: A review and empirical analysis,” *Energy Policy*, vol. 109, pp. 230–248, Oct. 2017.
[29] C. Roe, S. Meliopoulos, R. Entriken, and S. Chhaya, “Simulated demand response of a residential energy management system,” in *IEEE 2011 EnergyTech*, 2011, pp. 1–6.
[30] Yuemin Ding and S. H. Hong, “A model of demand response energy management system in industrial facilities,” 2013, pp. 241–246.
[31] S. Mitra, I. E. Grossmann, J. M. Pinto, and N. Arora, “Optimal production planning under time-sensitive electricity prices for continuous power-intensive processes,” *Comput. Chem. Eng.*, vol. 38, pp. 171–184, Mar. 2012.
[32] T. Samad and S. Kiliccote, “Smart grid technologies and applications for the industrial sector,” *Comput. Chem. Eng.*, vol. 47, pp. 76–84, Dec. 2012.
[33] H. Fraser, “The Importance of an Active Demand Side in the Electricity Industry,” *Electr. J.*, vol. 14, no. 9, pp. 52–73, Nov. 2001.
[34] G. Strbac, “Demand side management: Benefits and challenges,” *Energy Policy*, vol. 36, no. 12, pp. 4419–4426, Dec. 2008.
[35] S. Valero, M. Ortiz, C. Senabre, C. Alvarez, F. J. G. Franco, and A. Gabaldon, “Methods for customer and demand response policies selection in new electricity markets,” *IET Gener. Transm. Distrib.*, vol. 1, no. 1, p. 104, 2007.
[36] A. Mutanen, M. Ruska, S. Repo, and P. Jarventausta, “Customer Classification and Load Profiling Method for Distribution Systems,” *IEEE Trans. Power Deliv.*, vol. 26, no. 3, pp. 1755–1763, Jul. 2011.
[37] N. Mahmoudi-Kohan, M. K. Sheikh-El-Eslami, and S. M. Bidaki, “Improving WFA K-means Technique for Demand Response Programs Applications,” p. 5.
[38] A. J. Roscoe and G. Ault, “Supporting high penetrations of renewable generation via implementation of real-time electricity pricing and demand response,” *IET Renew. Power Gener.*, vol. 4, no. 4, p. 369, 2010.
[39] “Energy Interoperation Version 1.0.” [Online]. Available: http://docs.oasis-open.org/energyinterop/ei/v1.0/energyinterop-v1.0.html. [Accessed: 07-May-2018].

[40] "Extensible Markup Language (XML) 1.0 (Fifth Edition)." [Online]. Available: https://www.w3.org/TR/2008/REC-xml-20081126/. [Accessed: 07-May-2018].
[41] "OASIS Web Services Calendar (WS-Calendar) TC | OASIS." [Online]. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-calendar. [Accessed: 07-May-2018].
[42] "Energy Market Information Exchange (EMIX) Version 1.0." [Online]. Available: http://docs.oasis-open.org/emix/emix/v1.0/emix-v1.0.html. [Accessed: 07-May-2018].
[43] M. Piette *et al.*, "Open Automated Demand Response Communications Specification (Version 1.0)," 2009. .
[44] "Home." [Online]. Available: http://www.openadr.org/. [Accessed: 15-May-2018].
[45] "OpenADR - Nestfield." [Online]. Available: https://sites.google.com/a/nestfield.co.kr/nestfield/home_e/openadr. [Accessed: 15-May-2018].
[46] "OpenADR 2.0 Profile Specification B Profile." .
[47] Girish Ghatikar and R. Bienert, "Smart Grid Standards and Systems Interoperability: A Precedent with OpenADR," Unpublished, 2011.
[48] "Smart Energy Profile 2.0 Application Protocol Standard," p. 339, 2013.
[49] B. Ash and R. Simpson, "IEEE 2030.5: Relationship to other Smart Grid Standards, Testing and Certification," p. 73.
[50] "96/05639 BACnet. A data communication protocol for building automation and control networks," *Fuel Energy Abstr.*, vol. 37, no. 5, p. 390, Sep. 1996.
[51] "The Language of BACnet." [Online]. Available: http://www.bacnet.org/Bibliography/ES-7-96/ES-7-96.htm. [Accessed: 15-May-2018].
[52] V. Boed, *Networking and Integration of Facilities Automation Systems*. CRC Press, 1999.
[53] F. Luo, K. Q. Xiong, B. Li, M. Zhong, and H. G. Yan, "Extending IEC 61968 to Support Wide Area Demand Response Program," *Appl. Mech. Mater.*, vol. 701–702, pp. 1337–1342, Dec. 2014.
[54] C. Scott, "Comparison of Demand Response Communication Protocols," *Tech. Rep.*, p. 21, 2011.
[55] P. Simoneau, "The OSI Model: Understanding the Seven Layers of Computer Networks," p. 11.
[56] P. J. Leach, T. Berners-Lee, J. C. Mogul, L. Masinter, R. T. Fielding, and J. Gettys, "Hypertext Transfer Protocol -- HTTP/1.1." [Online]. Available: https://tools.ietf.org/html/rfc2616. [Accessed: 22-May-2018].
[57] L. Masinter, "Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0)." [Online]. Available: https://tools.ietf.org/html/rfc2324. [Accessed: 22-May-2018].
[58] M. Hauswirth and M. Jazayeri, "A Component and Communication Model for Push Systems?," p. 19.
[59] M. Shariff, A. Bhandari, P. Majmudar, and V. Choudhary, "Alfresco 3 Web Content Management," p. 32.
[60] M. M. HOSSAIN, "PORTLET-BASED PRESENTATION OF ENERGY KPIS IN SOA- ENABLED MANUFACTURING FACILITIES TARGETING HOLISTIC ENERGY MANAGEMENT," p. 95.
[61] D. Schachinger, S. Gaida, and W. Kastner, "Smart grid communication at the interface of customer buildings with focus on demand response," 2015, pp. 368–373.
[62] A. T. McKane *et al.*, "Opportunities, Barriers and Actions for Industrial Demand Response in California," LBNL-1335E, 945364, Jan. 2008.
[63] U. Herberg, D. Mashima, J. G. Jetcheva, and S. Mirzazad-Barijough, "OpenADR 2.0 deployment architectures: Options and implications," 2014, pp. 782–787.

[64] S. H. Hong, “Development of a BACnet-ZigBee gateway for demand response in buildings,” 2013, pp. 19–23.
[65] S. Son, H. Lee, and B.-T. Lee, “CoAP-based lightweight information exchange technique for demand response,” 2017, pp. 973–975.
[66] A. Keranen, M. Ersue, and C. Bormann, “Terminology for Constrained-Node Networks.” [Online]. Available: https://tools.ietf.org/html/rfc7228. [Accessed: 22-May-2018].
[67] T. Sato *et al.*, *Smart Grid Standards : Specifications, Requirements, and Technologies*. Singapore, SINGAPORE: John Wiley & Sons, Incorporated, 2015.
[68] W. T. ZHANG Jing, “Research on Technical Standardization for Electric Demand Response,” *Proceeding CSEE*, vol. 34, no. 22, pp. 3623–3629, 2014.
[69] M. A. Piette, G. Ghatikar, S. Kiliccote, D. Watson, E. Koch, and D. Hennage, “Design and Operation of an Open, Interoperable Automated Demand Response Infrastructure for Commercial Buildings,” *J. Comput. Inf. Sci. Eng.*, vol. 9, no. 2, p. 021004, 2009.
[70] “Introduction to Liferay Development - Liferay 7.0 - Liferay Developer Network.” [Online]. Available: https://dev.liferay.com/fi/develop/tutorials/-/knowledge_base/7-0/introduction-to-liferay-development. [Accessed: 11-Jul-2018].
[71] “Maven Repository: com.liferay.faces » liferay-faces-bridge-impl » 4.2.5-ga6.” [Online]. Available: https://mvnrepository.com/artifact/com.liferay.faces/liferay-faces-bridge-impl/4.2.5-ga6. [Accessed: 11-Jul-2018].
[72] D. A. Lobov and J. L. M. Lastra, “WAEL MOHAMMED ENCAPSULATION OF MES FUNCTIONALITIES AS RESTFUL WEB SERVICES FOR KNOWLEDGE-DRIVEN MANUFACTURING SYS- TEMS,” p. 65.

APPENDIX A – EMIX CLASSES AND MESSAGE SCHEMA

This XML file does not appear to have any style information associated with it. The document tree is shown below [42]


```
<!--

     Energy Market Information Exchange (EMIX) Version 1.0
     Committee Specification 02
     11 January 2012
     Copyright (c) OASIS Open 2012.  All Rights Reserved.
     Source: http://docs.oasis-open.org/emix/emix/v1.0/cs02/xsd/
-->
<!--
 emix.xsd
Schema agency:    OASIS EMIX TC
Schema version:    1.0
Schema date:       17 November 2011

Set includes:
    EMIX, EMIX-Termss, EMIX-Warrants (emix)
    Power, Power-Contracts, Power-Quality (power)
    Resource (resource)
-->
<!--  1.0 EMIX: Energy Market Information Exchange -->
<xs:schema xmlns:emix="http://docs.oasis-
open.org/ns/emix/2011/06" xmlns:scale="http://docs.oasis-
open.org/ns/emix/2011/06/siscale" xmlns:xcal="urn:ietf:params:xml:ns:ic
alendar-2.0" xmlns:clm5ISO42173A="urn:un:unece:uncefact:codelist:stand-
ard:5:ISO42173A:2010-04-07" xmlns:gml="http://www.open-
gis.net/gml/3.2" xmlns:gmlsf="http://www.open-
gis.net/gmlsf/2.0" xmlns:xs="http://www.w3.org/2001/XMLSchema" target-
Namespace="http://docs.oasis-open.org/ns/emix/2011/06" element-
FormDefault="qualified"attributeFormDefault="unqualified">
<xs:include schemaLocation="emix-terms.xsd"/>
<xs:include schemaLocation="emix-warrants.xsd"/>
<xs:import namespace="urn:un:unece:uncefact:codelist:stand-
ard:5:ISO42173A:2010-04-07" schemaLoca-
tion="http://www.unece.org/uncefact/codelist/standard/ISO_ISO3AlphaCur-
rencyCode_20100407.xsd"/>
<xs:import namespace="http://www.opengis.net/gml/3.2" schemaLoca-
tion="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
<xs:import namespace="http://www.opengis.net/gmlsf/2.0" schemaLoca-
tion="http://schemas.opengis.net/gmlsfProfile/2.0/gmlsfLevels.xsd"/>
<xs:import namespace="urn:ietf:params:xml:ns:icalendar-2.0" schemaLoca-
tion="http://docs.oasis-open.org/ws-calendar/ws-calendar-
spec/v1.0/cs01/xsd/iCalendar-wscal-extensions.xsd"/>
<xs:import namespace="urn:ietf:params:xml:ns:icalendar-2.0" schemaLoca-
tion="http://docs.oasis-open.org/ws-calendar/ws-calendar-
spec/v1.0/cs01/xsd/iCalendar.xsd"/>
<xs:import namespace="urn:ietf:params:xml:ns:icalendar-2.0" schemaLoca-
tion="http://docs.oasis-open.org/ws-calendar/ws-calendar-
spec/v1.0/cs01/xsd/iCalendar-availability-extension.xsd"/>
```

```
<xs:import namespace="urn:ietf:params:xml:ns:icalendar-2.0" schemaLoca-
tion="http://docs.oasis-open.org/ws-calendar/ws-calendar-
spec/v1.0/cs01/xsd/iCalendar-valtypes.xsd"/>
<!--  1.0 Core EMIX objects -->
<xs:annotation>
<xs:appinfo source="http://schemas.opengis.net/gmlsfPro-
file/2.0/gmlsfLevels.xsd">
<gmlsf:ComplianceLevel>0</gmlsf:ComplianceLevel>
</xs:appinfo>
</xs:annotation>
<!--  1.1 EMIX Product  -->
<xs:element name="product" type="emix:ProductType">
<xs:annotation>
<xs:documentation>
Emix Product, .i.e., a Product Description applied to a schedule.
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:complexType name="ProductType" mixed="false">
<xs:annotation>
<xs:documentation>
EMIX Product Type, i.e. a Product Description applied to a Schedule
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="emix:EmixBaseType">
<xs:sequence>
<xs:element ref="emix:transactiveState"/>
<xs:element ref="emix:expirationDate" minOccurs="0" maxOccurs="1"/>
<xs:element ref="emix:integralOnly" minOccurs="0" maxOccurs="1">
<xs:annotation>
<xs:documentation>
As part of a Tender, indicates element must be accepted in full or rejected
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element ref="emix:currency" minOccurs="0" maxOccurs="1"/>
<xs:element ref="emix:marketContext"/>
<xs:element ref="emix:side"/>
<xs:element ref="emix:terms" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--  1.2 EMIX Option  -->
<xs:element name="emixOption" type="emix:EmixOptionType">
<xs:annotation>
<xs:documentation>Option to buy an Emix Product</xs:documentation>
</xs:annotation>
</xs:element>
<xs:complexType name="EmixOptionType" mixed="false">
<xs:complexContent>
<xs:extension base="emix:EmixBaseType">
<xs:sequence>
<xs:element ref="emix:transactiveState"/>
<xs:element ref="emix:expirationDate" minOccurs="0" maxOccurs="1"/>
<xs:element ref="emix:integralOnly" minOccurs="0" maxOccurs="1">
<xs:annotation>
<xs:documentation>
If true, each Option Call must be for the full amount specified
</xs:documentation>
```

```
</xs:annotation>
</xs:element>
<xs:element name="optionHolderSide" type="emix:SideType"/>
<xs:element name="optionPremium" type="emix:PriceType"/>
<xs:element name="optionStrikePrice" type="emix:PriceType"/>
<xs:element ref="emix:optionType"/>
<xs:element ref="emix:side"/>
<xs:element ref="emix:marketContext"/>
<xs:element ref="emix:currency" minOccurs="0" maxOccurs="1"/>
<xs:element ref="emix:terms" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--  1.4 Delivery  -->
<xs:element name="delivery" type="emix:DeliveryType"/>
<xs:complexType name="DeliveryType" mixed="false">
<xs:annotation>
<xs:documentation>
Receipt / Report of Product Delivery. Injection flag is true for adding
product to market supply, false for taking from market.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="emix:EmixBaseType">
<xs:sequence>
<xs:element name="injection" type="xs:boolean"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--  2.0 EMIX Components -->
<!--  2.1 Envelope  -->
<xs:element name="envelopeContents" type="emix:EnvelopeContentsType"/>
<xs:complexType name="EnvelopeContentsType">
<xs:sequence>
<xs:element ref="emix:warrants" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>

<!--  8.0 Supporting Information Structures -->
<!--  8.2 Market defintions  -->
<!--  8.2.1 Market Context  -->

<xs:element name="marketContext" type="emix:MarketContextType"/>
<xs:simpleType name="MarketContextType">
<xs:restriction base="xs:anyURI"/>
</xs:simpleType>

<!--  8.2.2 Transactive State  -->

<xs:element name="transactiveState" type="emix:TransactiveStateType"/>
<xs:simpleType name="TransactiveStateType">
<xs:restriction base="xs:string">
<xs:enumeration value="indicationOfInterest"/>
<xs:enumeration value="tender"/>
<xs:enumeration value="transaction"/>
<xs:enumeration value="exercise"/>
<xs:enumeration value="delivery"/>
<xs:enumeration value="transportCommitment"/>
<xs:enumeration value="publication"/>
```

```
</xs:restriction>
</xs:simpleType>

<!--  8.2.3 Currency  -->

<xs:element name="currency" type="clm5ISO42173A:ISO3AlphaCurrencyCo-
deContentType">
<xs:annotation>
<xs:documentation>Currency codes coming from UN CEFACT schemas</xs:docu-
mentation>
</xs:annotation>
</xs:element>

<!--  8.2.4 Enumeration for Side  -->

<xs:element name="side" type="emix:SideType"/>
<xs:simpleType name="SideType">
<xs:restriction base="xs:string">
<xs:enumeration value="buy"/>
<xs:enumeration value="sell"/>
</xs:restriction>
</xs:simpleType>

<!--  8.3 Price  -->

<xs:element name="priceBase" type="emix:PriceBaseType" abstract="true">
<xs:annotation>
<xs:documentation>Abstract base for EMIX Prices</xs:documentation>
</xs:annotation>
</xs:element>
<xs:complexType name="PriceBaseType" abstract="true">
<xs:annotation>
<xs:documentation>Type of Abstract base for EMIX Prices</xs:documenta-
tion>
</xs:annotation>
</xs:complexType>

<!--  8.3.1 Absolute Price   -->

<xs:element name="price" type="emix:PriceType" substitu-
tionGroup="emix:priceBase"/>
<xs:complexType name="PriceType" mixed="false">
<xs:annotation>
<xs:documentation>Simple Price</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="emix:PriceBaseType">
<xs:sequence>
<xs:element ref="emix:value" minOccurs="1" maxOccurs="1"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<!--  8.3.2 Multiplier Price - multiplier on base amount   -->

<xs:element name="priceMultiplier" type="emix:PriceMultiplierType" sub-
stitutionGroup="emix:priceBase"/>
<xs:complexType name="PriceMultiplierType" mixed="false">
<xs:annotation>
<xs:documentation>
```

```
Multiplier times market price, 1 for same as market
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="emix:PriceBaseType">
<xs:sequence>
<xs:element name="multiplier" type="xs:float" minOccurs="1" max-
Occurs="1"/>
<xs:element ref="emix:marketContext" minOccurs="0" maxOccurs="1">
<xs:annotation>
<xs:documentation>
Market Context for base price. If blank, Market Context from hosting
artifact.
</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<!--
 8.3.4 Price Offset (additive or subtractive) over base amount
-->

<xs:element name="priceRelative" type="emix:PriceRelativeType" substi-
tutionGroup="emix:priceBase"/>
<xs:complexType name="PriceRelativeType" mixed="false">
<xs:annotation>
<xs:documentation>
Price Relative is a fixed charge (positive or negative) apllied to base
price
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="emix:PriceBaseType">
<xs:sequence>
<xs:element ref="emix:value" minOccurs="1" maxOccurs="1"/>
<xs:element ref="emix:marketContext" minOccurs="0" maxOccurs="1">
<xs:annotation>
<xs:documentation>
Market Context for base price. If blank, Market Context from hosting
artifact.
</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<!--  8.3.6 Simple Price   -->

<xs:element name="value" type="emix:ValueType"/>
<xs:simpleType name="ValueType">
<xs:restriction base="xs:decimal"/>
</xs:simpleType>

<!--  8.5 Quantity  -->

<xs:element name="integralOnly" type="emix:IntegralOnlyType"/>
```

```
<xs:simpleType name="IntegralOnlyType">
<xs:annotation>
<xs:documentation>
Integral Only is an indication that the element described is [tendered]
as an all or nothing product. It may apply to an (amount, response, ramp)
that is all (true) or nothing (false)
</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:boolean"/>
</xs:simpleType>
<xs:element name="autonomous" type="emix:AutonomousType"/>
<xs:simpleType name="AutonomousType">
<xs:annotation>
<xs:documentation>
An autonomous resource or service (true) is able to respond or maintain
service independently. A non autonomous service (false) must await dis-
patch.
</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:boolean"/>
</xs:simpleType>

<!--  8.7 Enumeration for Option Types  -->

<xs:element name="optionType" type="emix:OptionTypeType"/>
<xs:simpleType name="OptionTypeType">
<xs:union memberTypes="emix:OptionTypeEnumeratedType emix:EmixExtension-
Type"/>
</xs:simpleType>
<xs:simpleType name="OptionTypeEnumeratedType">
<xs:annotation>
<xs:documentation>Enumerated Option Types</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string"/>
</xs:simpleType>

<!--9.1 Abstract EMIX Base(product applied to a schedule)-->

<xs:element name="emixBase" type="emix:EmixBaseType"/>
<xs:complexType name="EmixBaseType" abstract="true">
<xs:annotation>
<xs:documentation> iCalendar-derived object to host EMIX ele-
ments</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="xcal:VcalendarType">
<xs:sequence>
<xs:element ref="emix:uid" minOccurs="1" maxOccurs="1"/>
<xs:element ref="emix:envelopeContents" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<!--  9.2 Abstract Product  Description  -->

<xs:element name="productDescription" type="emix:ProductDescription-
Type" abstract="true" substitutionGroup="xcal:artifactBase"/>
<xs:complexType name="ProductDescriptionType" abstract="true">
<xs:annotation>
<xs:documentation>
```

```
In EMIX, the Product Description is placed in the Interval or Gluon
attachment. The respective product schemas extend this abstract class.
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="xcal:ArtifactBaseType"/>
</xs:complexContent>
</xs:complexType>
<!--  9.3 Interfaces   -->
<xs:element name="serviceArea" type="emix:ServiceAreaType" substitu-
tionGroup="emix:emixInterface"/>
<xs:complexType name="ServiceAreaType">
<xs:annotation>
<xs:documentation>
The Service Area is the geographic region that is affected by the EMIX
market condition
</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="emix:EmixInterfaceType">
<xs:sequence>
<xs:element ref="gml:AbstractFeature" minOccurs="1" maxOccurs="un-
bounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="emixInterface" type="emix:EmixInterfaceType"/>
<xs:complexType name="EmixInterfaceType" abstract="true" mixed="false">
<xs:annotation>
<xs:documentation>
Abstract base class for the interfaces for EMIX Product delivery, meas-
urement, and/or pricing
</xs:documentation>
</xs:annotation>
</xs:complexType>
<!--  9.4 emix Measuerement  -->
<xs:element name="measurement" type="emix:MeasurementType" substitu-
tionGroup="emix:productDescription"/>
<xs:complexType name="MeasurementType">
<xs:annotation>
<xs:documentation>Type of Measurement</xs:documentation>
</xs:annotation>
<xs:complexContent>
<xs:extension base="emix:ProductDescriptionType">
<xs:sequence>
<xs:element ref="emix:quantity"/>
<xs:element ref="emix:itemBase"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!--  9.5 Granularity  -->
<xs:element name="emixGranularity" type="emix:EmixGranularityType"/>
<xs:complexType name="EmixGranularityType" mixed="false">
<xs:annotation>
<xs:documentation>
Abstract base class used for granularity of market indications of interest
and tenders
</xs:documentation>
</xs:annotation>
```

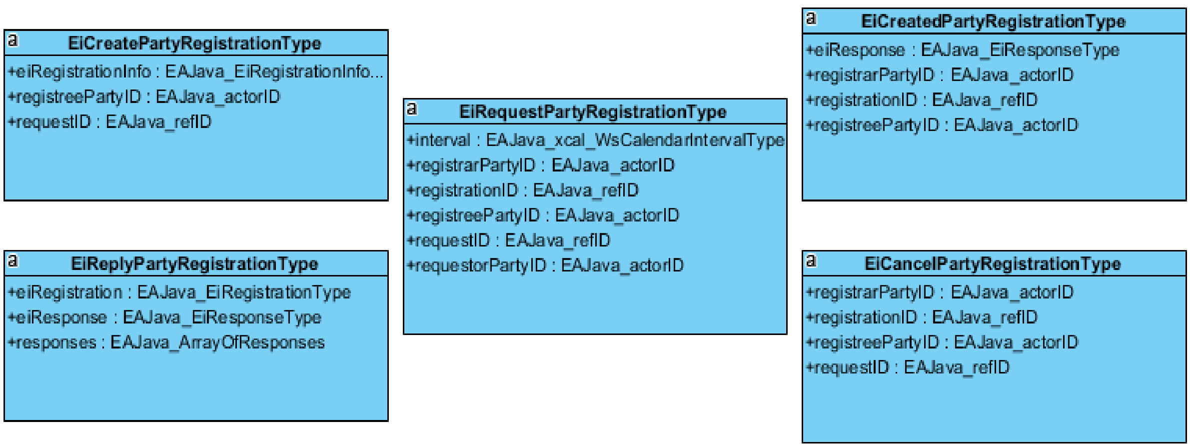
```
<xs:sequence>
<xs:element ref="emix:quantity"/>
<xs:element ref="emix:itemBase"/>
</xs:sequence>
</xs:complexType>
<!--  9.5 Item Base  -->
<xs:element name="itemBase" type="emix:ItemBaseType" abstract="true"/>
<xs:complexType name="ItemBaseType" abstract="true" mixed="false">
<xs:annotation>
<xs:documentation>
Abstract base type for units for EMIX Product delivery, measurement, and
warrants. Item as in PO Item, Requisition Item, Invoice Item, Lading
Item. Item does not include Quantity or Price, because a single product
description or transaction may have multiple quantities or prices asso-
ciated with a single item.
</xs:documentation>
</xs:annotation>
</xs:complexType>
<!--  9.8 Simple Types   -->
<xs:element name="uid" type="emix:UidType"/>
<xs:simpleType name="UidType">
<xs:annotation>
<xs:documentation>
A unique identifier for an EMIX Type. Different markets and specifications
that use EMIX may have their own rules for specifying an UID.
</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:element name="quantity" type="emix:QuantityType"/>
<xs:simpleType name="QuantityType">
<xs:annotation>
<xs:documentation>Base type for all quanties in EMIX.</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:float"/>
</xs:simpleType>
<xs:element name="expirationDate" type="xcal:DateTimeType">
<xs:annotation>
<xs:documentation>
Expiration date for tenders and similar related information.
</xs:documentation>
</xs:annotation>
</xs:element>
<!--  9.9 Extension Type   -->
<xs:simpleType name="EmixExtensionType">
<xs:annotation>
<xs:documentation>
Pattern used for extending string enumeration, where allowed
</xs:documentation>
</xs:annotation>
<xs:restriction base="xs:string">
<xs:pattern value="x-\S.*"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>
```

APPENDIX B – EIREGISTERATION AND EIPARTY SCHEMA'S

The corresponding XSD (XML Schema Definition) for EiRegisterationType and EiParty is given below:

```
<xs:element name="eiRegistration" type="ei:EiRegistrationType"/>
<xs:complexType name="EiRegistrationType" abstract="true">
    <xs:sequence>
        <xs:element ref="ei:registreePartyID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="ei:registrarPartyID" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="ei:eiRegistrationInfo" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<!-- 5.2 Party-->
<!--    <xs:element name="registeredParty" type="ei:EiPartyType"/> -->
<xs:element name="eiParty" type="ei:EiPartyType"/>
<xs:complexType name="EiPartyType">
    <xs:sequence>
        <xs:element ref="ei:partyID" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="ei:partyName" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="ei:partyRole" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="ei:partyLocation" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
```

The Figure below depicts eiRegisteration operation payload. EiRequestPartyRegistrationType, EiReplyPartyRegistrationType, EiCreatePartyRegistrationType, EiCreatedPartyRegistrationType, EiCancelPartyRegistrationType are the information model communicated sequentially between the DR participants during registration.



The corresponding XSD (XML Schema Definition) for eiRegisteration operation payload is given below:

eiCreatePartyRegisteration:

```
<xs:element name="eiCreatePartyRegistration" type="pyld:EiCreatePartyRegistra-
tionType"/>
<xs:complexType name="EiCreatePartyRegistrationType">
      <xs:annotation>
            <xs:documentation>Used to create and send a Party Registration re-
            quest.</xs:documentation>
      </xs:annotation>
      <xs:sequence>
            <xs:element ref="pyld:requestID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:registreePartyID" minOccurs="0" max-
      Occurs="1"/>
            <xs:element ref="ei:eiRegistrationInfo" minOccurs="1" max-
      Occurs="1"/>
      </xs:sequence>
</xs:complexType>
```

eiCreatedPartyRegisteration:

```
<xs:element name="eiCreatedPartyRegistration" type="pyld:EiCreatedPartyRegis-
trationType"/>
<xs:complexType name="EiCreatedPartyRegistrationType">
      <xs:sequence>
            <xs:element ref="ei:eiResponse" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:registrarPartyID" minOccurs="0" max-
            Occurs="1"/>
            <xs:element ref="ei:registrationID" minOccurs="0" maxOccurs="1"/>
            <xs:element ref="ei:registreePartyID" minOccurs="0" max-
            Occurs="1"/>
      </xs:sequence>
</xs:complexType>
```

eiRequestPartyRegisteration:

```
<xs:element name="eiRequestPartyRegistration" type="pyld:EiRequestPartyRegis-
trationType"/>
<xs:annotation>
      <xs:documentation>Request for information about extant registra-
      tions.</xs:documentation>
</xs:annotation>
<xs:complexType name="EiRequestPartyRegistrationType">
      <xs:sequence>
            <xs:element ref="pyld:requestID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:registreePartyID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:registrarPartyID" minOccurs="0" maxOccurs="1"/>
            <xs:element ref="ei:requestorPartyID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:registrationID" minOccurs="0" maxOccurs="un-
            bounded"/>
            <xs:element ref="xcal:interval" minOccurs="0" maxOccurs="1">
            <xs:annotation>
                  <xs:documentation>If present, limits range of request to reg-
                  istrations within Interval.</xs:documentation>
            </xs:annotation>
            </xs:element>
      </xs:sequence>
</xs:complexType>
```

eiReplyPartyRegisteration:

```
<xs:element name="eiReplyPartyRegistration" type="pyld:EiReplyPartyRegistra-
tionType"/>
<xs:complexType name="EiReplyPartyRegistrationType">
      <xs:sequence>
            <xs:element ref="ei:eiRegistration" minOccurs="0" maxOccurs="un-
            bounded"/>
            <xs:element ref="ei:eiResponse" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:responses" minOccurs="0" maxOccurs="1"/>
            </xs:sequence>
</xs:complexType>
```

eiCancelPartyRegisteration:

```
<xs:element name="eiCancelPartyRegistration" type="pyld:EiCancelPartyRegistra-
tionType"/>
<xs:complexType name="EiCancelPartyRegistrationType">
<xs:annotation>
      <xs:documentation>Used to cancel one or more Party Registrations.</xs:doc-
      umentation>
</xs:annotation>
      <xs:sequence>
            <xs:element ref="pyld:requestID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:registreePartyID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:registrarPartyID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:registrationID" minOccurs="1" maxOccurs="un-
            bounded"/>
      </xs:sequence>
</xs:complexType>
```

eiCanceledPartyRegisteration:

```
<xs:element name="eiCanceledPartyRegistration" type="pyld:EiCanceledPartyReg-
istrationType"/>
<xs:complexType name="EiCanceledPartyRegistrationType">
      <xs:sequence>
            <xs:element ref="ei:respondingPartyID" minOccurs="1" max-
            Occurs="1"/>
            <xs:element ref="ei:registreePartyID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:registrarPartyID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:eiResponse" minOccurs="0" maxOccurs="un-
            bounded"/>
            <xs:element ref="ei:responses" minOccurs="0" maxOccurs="un-
            bounded"/>
      </xs:sequence>
</xs:complexType>
```

APPENDIX C – EIQUOTE AND EMIXBASE TYPE SCHEMA'S

The corresponding XSD (XML Schema Definition) for EiQuoteType and EmixBaseType is given below in Figures:

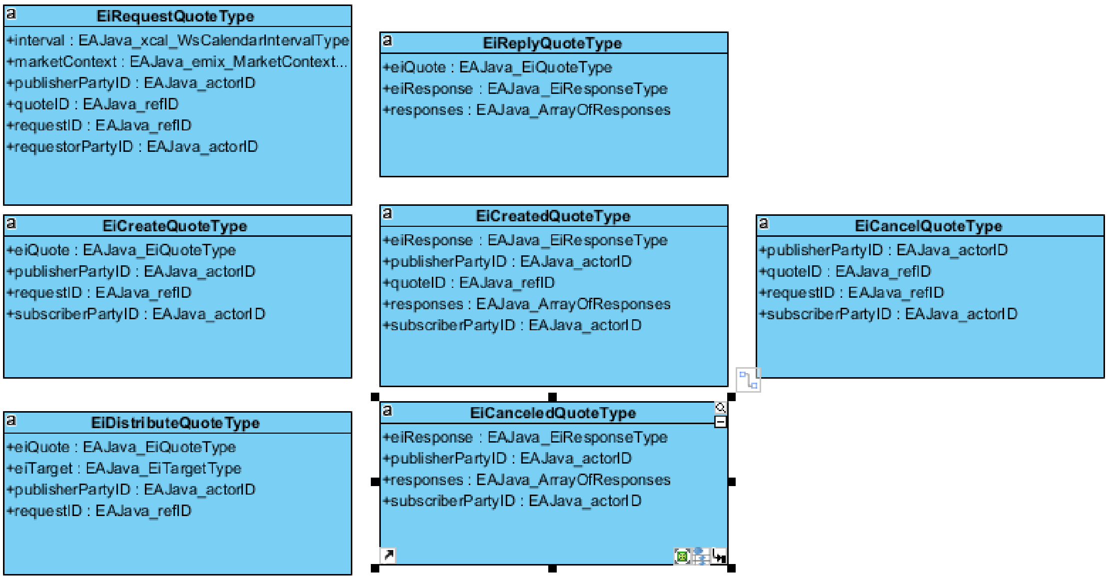
```
<xs:element name="eiQuote" type="ei:EiQuoteType"/>
<xs:complexType name="EiQuoteType">
    <xs:annotation>
        <xs:documentation>Day-ahead electricity price</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element ref="ei:quoteID" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="emix:emixBase" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

eiQuote message schema

```
            <xs:extension base="emix:EmixBaseType">
                <xs:sequence>
                    <xs:element ref="emix:transactiveState"/>
                    <xs:element ref="emix:expirationDate" minOccurs="0" maxOccurs="1"/>
                    <xs:element ref="emix:integralOnly" minOccurs="0" maxOccurs="1"/>
                    <xs:element ref="emix:currency" minOccurs="0" maxOccurs="1"/>
                    <xs:element ref="emix:marketContext"/>
                    <xs:element ref="emix:side"/>
                    <xs:element ref="emix:terms" minOccurs="0" maxOccurs="1"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
```

EMIXBaseType message schema

The Figure below depicts eiQuote operation payload. They are the day-ahead electricity price information model communicated between the utility and facility.



eiQuote operation payload

The corresponding XSD (XML Schema Definition) for eiQuote operation payload is given below:

eiCreateQuote:

```
<xs:element name="eiCreateQuote" type="pyld:EiCreateQuoteType"/>
<xs:complexType name="EiCreateQuoteType">
<xs:annotation>
      <xs:documentation>Used to create and send a Quote.</xs:documentation>
</xs:annotation>
      <xs:sequence>
            <xs:element ref="pyld:requestID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:publisherPartyID" minOccurs="1" maxOccurs="1"/>
            <xs:element    ref="ei:subscriberPartyID"    minOccurs="1"    max-
            Occurs="1"/>
            <xs:element ref="ei:eiQuote" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
</xs:complexType>
```

eiCreatedQuote:

```
<xs:element name="eiCreatedQuote" type="pyld:EiCreatedQuoteType"/>
<xs:complexType name="EiCreatedQuoteType">
      <xs:sequence>
            <xs:element ref="ei:eiResponse" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:responses" minOccurs="0" maxOccurs="1"/>
            <xs:element    ref="ei:subscriberPartyID"    minOccurs="1"    max-
            Occurs="1"/>
            <xs:element ref="ei:publisherPartyID" minOccurs="0" maxOccurs="1"/>
            <xs:element ref="ei:quoteID" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
</xs:complexType>
```

eiCancelQuote:

```
<xs:element name="eiCancelQuote" type="pyld:EiCancelQuoteType"/>
<xs:complexType name="EiCancelQuoteType">
<xs:annotation>
      <xs:documentation>Used to cancel a Quote.</xs:documentation>
</xs:annotation>
      <xs:sequence>
            <xs:element ref="pyld:requestID" minOccurs="1" maxOccurs="1"/>
            <xs:element    ref="ei:subscriberPartyID"    minOccurs="0"    max-
            Occurs="1"/>
            <xs:element ref="ei:publisherPartyID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:quoteID" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
</xs:complexType>
```

eiCanceledQuote:

```
<xs:element name="eiCanceledQuote" type="pyld:EiCanceledQuoteType"/>
<xs:complexType name="EiCanceledQuoteType">
      <xs:sequence>
            <xs:element ref="ei:eiResponse" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:responses" minOccurs="0" maxOccurs="1"/>
            <xs:element    ref="ei:subscriberPartyID"    minOccurs="0"    max-
            Occurs="1"/>
            <xs:element ref="ei:publisherPartyID" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
</xs:complexType>
```

eiRequestQuote:

```
<xs:element name="eiRequestQuote" type="pyld:EiRequestQuoteType"/>
<xs:complexType name="EiRequestQuoteType">
      <xs:sequence>
            <xs:element ref="pyld:requestID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:requestorPartyID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:publisherPartyID" minOccurs="1" maxOccurs="1"/>
            <xs:element  ref="emix:marketContext"  minOccurs="0"  maxOccurs="un-
            bounded"/>
            <xs:element ref="ei:quoteID" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="xcal:interval" minOccurs="0" maxOccurs="1">
      </xs:sequence>
</xs:complexType>
```

eiReplyQuote:

```
<xs:element name="eiReplyQuote" type="pyld:EiReplyQuoteType"/>
<xs:complexType name="EiReplyQuoteType">
      <xs:sequence>
            <xs:element ref="ei:eiResponse" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:responses" minOccurs="0" maxOccurs="1"/>
            <xs:element ref="ei:eiQuote" minOccurs="0" maxOccurs="unbounded"/>
```

```
      </xs:sequence>
</xs:complexType>
```

eiDistributeQuote:

```
<xs:element name="eiDistributeQuote" type="pyld:EiDistributeQuoteType"/>
<xs:complexType name="EiDistributeQuoteType">
<xs:annotation>
      <xs:documentation>For price distribution through broadcast, the VTN-VEN subscribe to the corresponding publisherID</xs:documentation>
</xs:annotation>
      <xs:sequence>
            <xs:element ref="pyld:requestID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:publisherPartyID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:eiTarget" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:eiQuote" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
</xs:complexType>
```

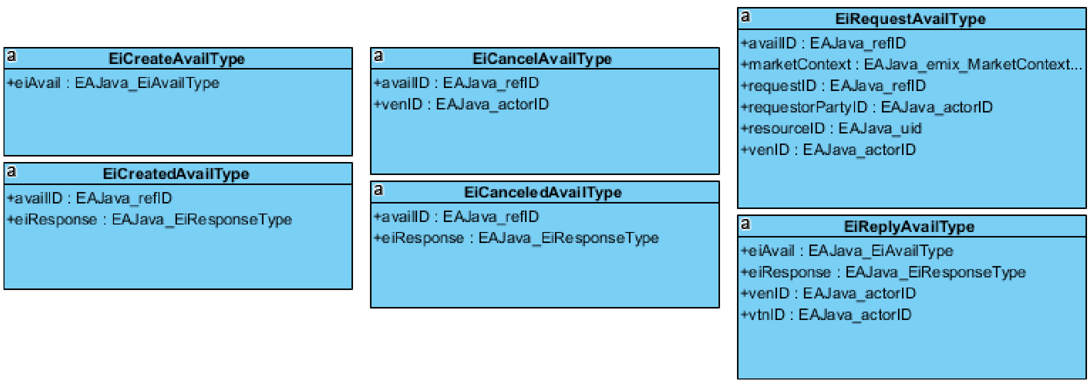
APPENDIX D – EIAVAILABLITY SCHEMA

The corresponding XSD (XML Schema Definition) for EiAvailType and EiAvailBehaviourType is given below in Figure below:

```
<!-- EiAvailType  -->
<xs:element name="eiAvail" type="ei:EiAvailType"/>
<xs:complexType name="EiAvailType">
    <xs:sequence>
        <xs:element ref="ei:availID" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="ei:venID" minOccurs="1" maxOccurs="unbounded"/>
        <xs:element ref="ei:resourceID" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="available" type="xcal:VavailabilityType" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="ei:createdDateTime" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="ei:eiAvailBehavior" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="emix:marketContext" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
<!-- EiAvailBehaviourType-->
<xs:element name="eiAvailBehavior" type="ei:EiAvailBehaviorType"/>
<xs:simpleType name="EiAvailBehaviorType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="accept"/>
        <xs:enumeration value="reject"/>
        <xs:enumeration value="restrict"/>
    </xs:restriction>
</xs:simpleType>
```

EiAvail and EiAvailBehaviour message schema

The Figure below depicts eiAvail operation payload. These are the actual messages passed between the VTN and VEN to share the details of resource availability in facility site.



eiAvailability operation payload

The corresponding XSD (XML Schema Definition) for eiAvailablity operation payload is given below:

eiCreateAvail

```
<xs:element name="eiCreateAvail" type="pyld:EiCreateAvailType"/>
<xs:complexType name="EiCreateAvailType">
      <xs:sequence>
            <xs:element ref="ei:eiAvail" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
</xs:complexType>
```

eiCreatedAvail

```
<xs:element name="eiCreatedAvail" type="pyld:EiCreatedAvailType"/>
<xs:complexType name="EiCreatedAvailType">
      <xs:sequence>
            <xs:element ref="ei:availID" minOccurs="0" maxOccurs="1"/>
            <xs:element ref="ei:eiResponse" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
</xs:complexType>
```

eiRequestAvail

```
<xs:element name="eiRequestAvail" type="pyld:EiRequestAvailType"/>
<xs:complexType name="EiRequestAvailType">
      <xs:sequence>
            <xs:element ref="pyld:requestID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:resourceID" minOccurs="0" maxOccurs="1"/>
            <xs:element ref="ei:requestorPartyID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:availID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="emix:marketContext" minOccurs="0" maxOccurs="un-
            bounded"/>
            <xs:element ref="ei:venID" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
</xs:complexType>
```

eiReplyAvail

```
<xs:element name="eiReplyAvail" type="pyld:EiReplyAvailType"/>
<xs:complexType name="EiReplyAvailType">
      <xs:sequence>
            <xs:element ref="ei:eiAvail" minOccurs="0" maxOccurs="1"/>
            <xs:element ref="ei:eiResponse" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:venID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:vtnID" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
</xs:complexType>
```

eiCancelAvail

```
<xs:element name="eiCancelAvail" type="pyld:EiCancelAvailType"/>
```

```
<xs:complexType name="EiCancelAvailType">
      <xs:sequence>
            <xs:element ref="ei:availID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:venID" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
</xs:complexType>
```

eiCanceledAvail

```
<xs:element name="eiCanceledAvail" type="pyld:EiCanceledAvailType"/>
<xs:complexType name="EiCanceledAvailType">
      <xs:sequence>
            <xs:element ref="ei:availID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:eiResponse" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
</xs:complexType>
```

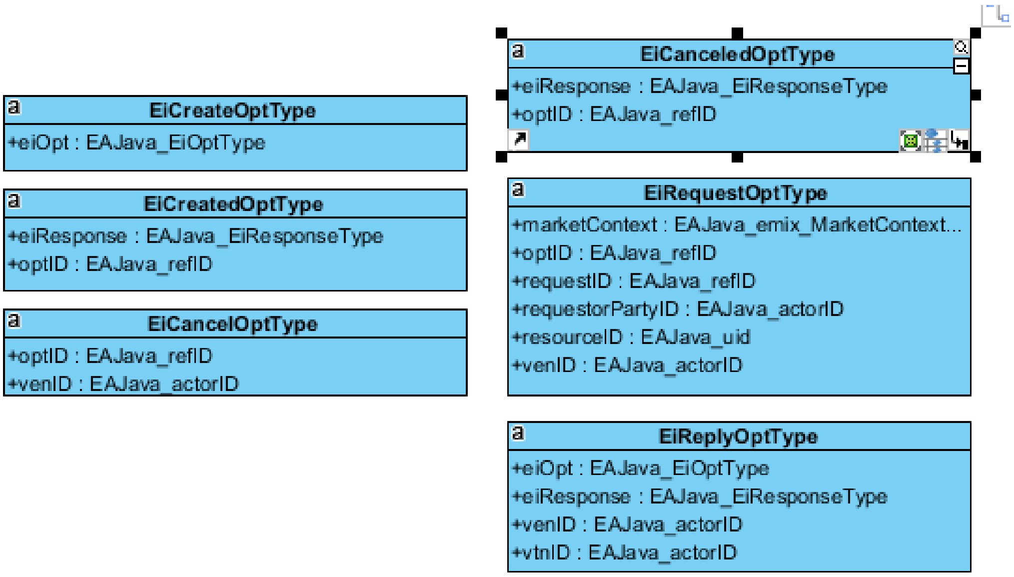
APPENDIX E – EIOVERRIDE SCHEMA

The corresponding XSD (XML Schema Definition) for EiOptType is given below:

```
<xs:element name="eiOpt" type="ei:EiOptType"/>
<xs:complexType name="EiOptType">
    <xs:sequence>
        <xs:element ref="ei:optID" maxOccurs="1">
        </xs:element>
        <xs:element ref="ei:optType" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="ei:optReason" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="emix:marketContext" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="ei:venID" minOccurs="1" maxOccurs="unbounded"/>
        <xs:element ref="xcal:vavailability" minOccurs="0"/>
        <xs:element ref="ei:eiMarketRuleSet" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="ei:createdDateTime" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute ref="ei:schemaVersion" use="optional"/>
</xs:complexType>
```

EiOverride message schema

The Figure below depicts eiOverride operation payload. These are the actual messages passed by VEN's to VTN's, when a VEN wants to override the actual availability service.



eiOverride operation payload

The corresponding XSD (XML Schema Definition) for eiOverride operation payload is given below:

eiCreateOpt

```
<xs:element name="eiCreateOpt" type="pyld:EiCreateOptType"/>
<xs:complexType name="EiCreateOptType">
      <xs:sequence>
            <xs:element ref="ei:eiOpt" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
</xs:complexType>
```

eiCreatedOpt

```
<xs:element name="eiCreatedOpt" type="pyld:EiCreatedOptType"/>
<xs:complexType name="EiCreatedOptType">
      <xs:sequence>
            <xs:element ref="ei:eiResponse" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:optID" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
</xs:complexType>
```

eiCancelOpt

```
<xs:element name="eiCancelOpt" type="pyld:EiCancelOptType"/>
<xs:complexType name="EiCancelOptType">
      <xs:sequence>
            <xs:element ref="ei:optID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:venID" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
</xs:complexType>
```

eiCanceledOpt

```
<xs:element name="eiCanceledOpt" type="pyld:EiCanceledOptType"/>
<xs:complexType name="EiCanceledOptType">
      <xs:sequence>
            <xs:element ref="ei:eiResponse" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:optID" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
</xs:complexType>
```

eiRequestOpt

```
<xs:element name="eiRequestOpt" type="pyld:EiRequestOptType"/>
<xs:complexType name="EiRequestOptType">
      <xs:sequence>
            <xs:element ref="pyld:requestID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:venID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="emix:marketContext" minOccurs="0" maxOccurs="1"/>
            <xs:element ref="ei:optID" minOccurs="1" maxOccurs="1"/>
```

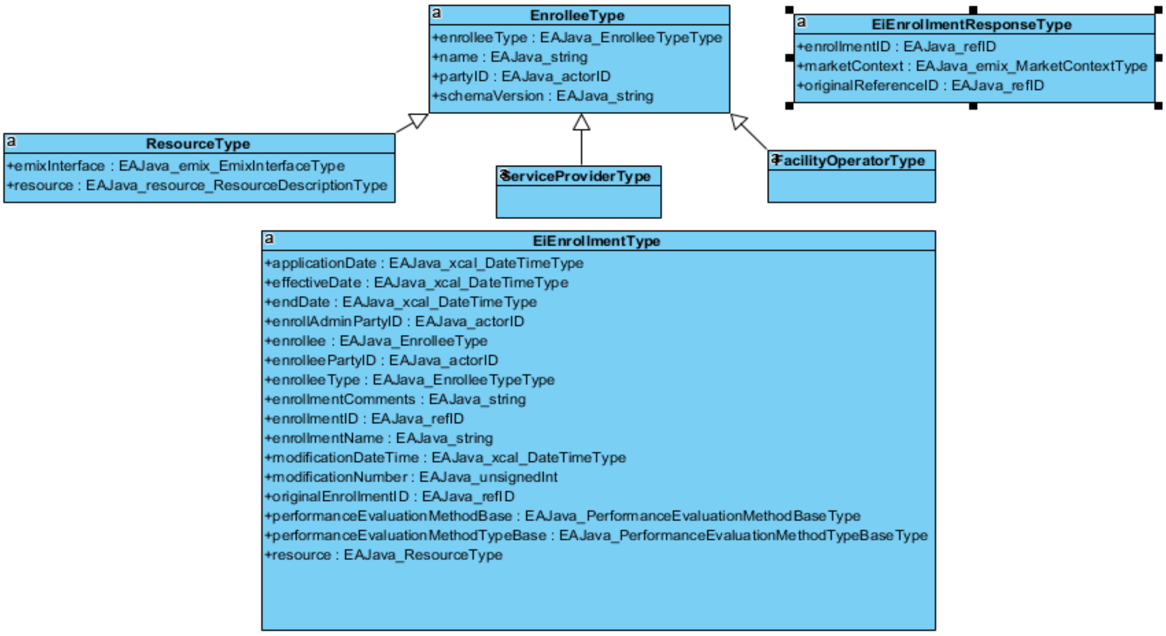
```
            <xs:element ref="ei:requestorPartyID" minOccurs="1" max
            Occurs="1"/>
            <xs:element ref="ei:resourceID" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
</xs:complexType>
```

eiReplyOpt

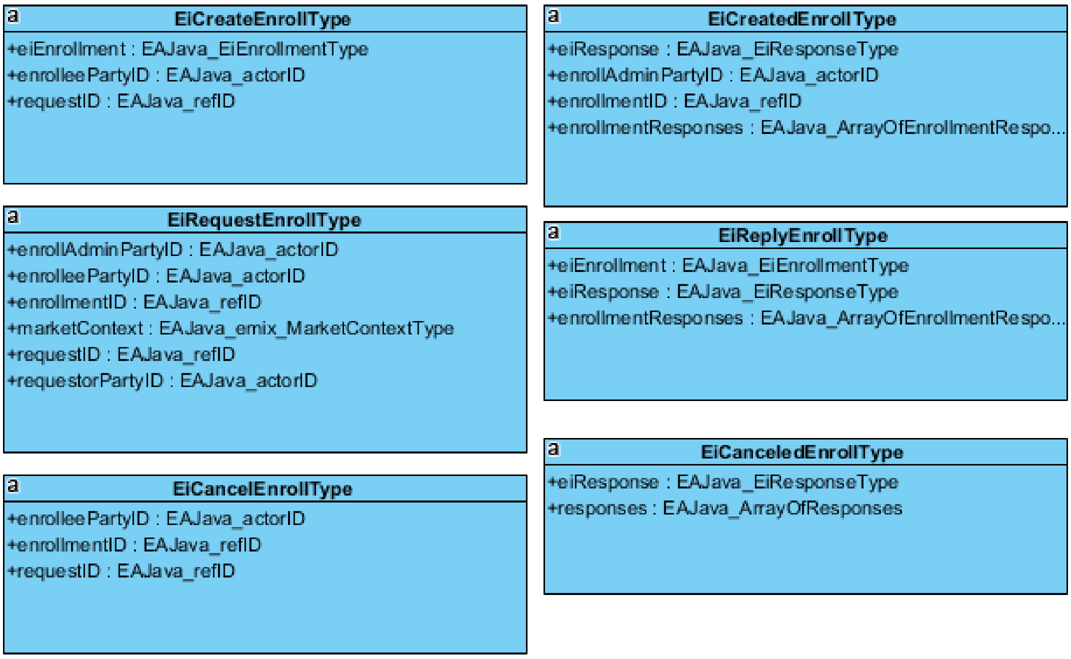
```
<xs:element name="eiReplyOpt" type="pyld:EiReplyOptType"/>
<xs:complexType name="EiReplyOptType">
      <xs:sequence>
            <xs:element ref="ei:eiResponse" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:venID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:vtnID" minOccurs="1" maxOccurs="1"/>
            <xs:element ref="ei:eiOpt" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
</xs:complexType>
```

APPENDIX F – EIENROLLEMENT SCHEMA

eiEnrollement Service



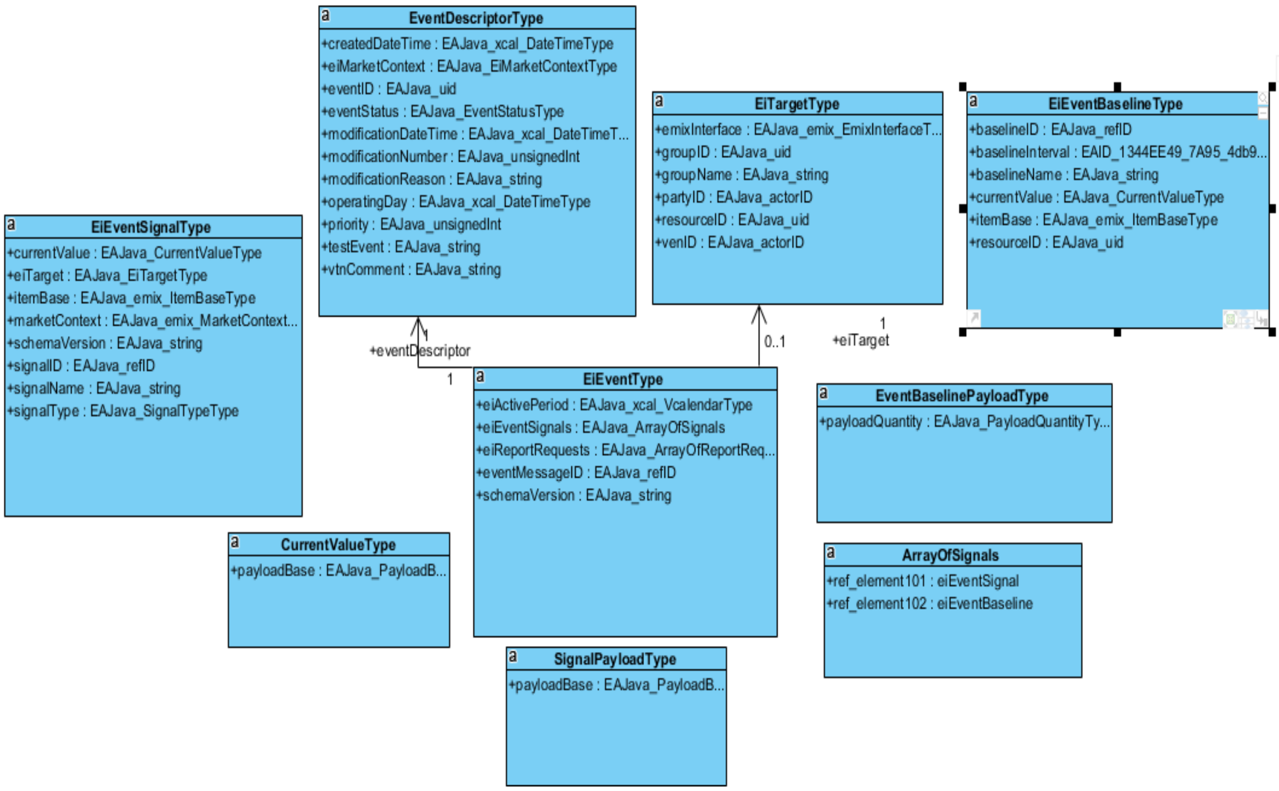
eiEnrollement service class diagram



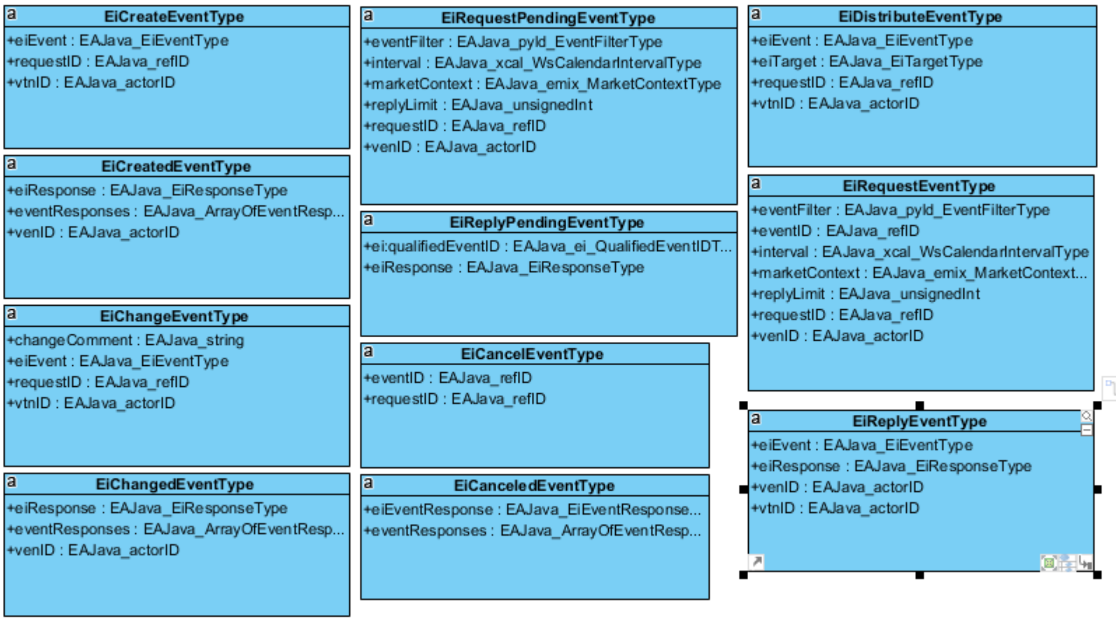
eiEnrollement operation payload

APPENDIX G – EIEVENT SCHEMA

eiEvent Service



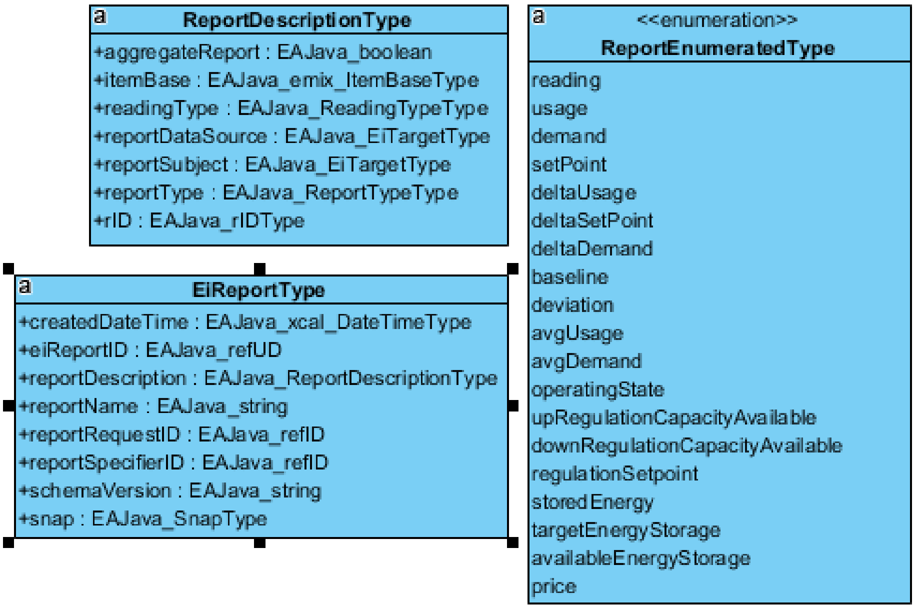
eiEvent class diagram



eiEvent operation payload

APPENDIX H – EIREPORT SCHEMA

eiReport Service



eiReport class diagram

eiReport operation payload